# Task Allocation Strategies in Multi-robot Environment

**A THESIS SUBMITTED FOR AWARD OF THE DEGREE OF DOCTOR OF PHILOSOPHY IN ENGINEERING**

**2009**

**By**

**Bibhuti Bhusan Choudhury**

**NATIONAL INSTITUTE OF TECHNOLOGY**
**ROURKELA**
**ORISSA, INDIA**

*Dedicated To*
*My*
*Parents*

*And*

*My*
*Teachers*

**NATIONAL INSTITUTE OF TECHNOLOGY**
**ROURKELA, INDIA**

**Dr.Bibhuti Bhusan Biswal**
**Professor**
**Department of Mechanical Engineering**
**NIT Rourkela**

# CERTIFICATE

**Date: July 13, 2009**

This is to certify that the thesis entitled **"Task Allocation Strategies in Multi-robot Environment",** being submitted by **Sri Bibhuti Bhusan Choudhury,** who got his name registered on 18.07.2006 (Roll No.50603003) to National Institute of Technology Rourkela for the award of the degree of **Doctor of Philosophy in Engineering** is a record of bonafide research work done by him under my supervision and guidance. The thesis satisfies the requirements of the regulations relating to the degree.

The work reported in this thesis has not been, to the best my knowledge, submitted to any other University or Institution for the award of any degree or diploma.

He bears a good moral character to the best of my knowledge and belief.

(Dr.B.B.Biswal)

# Acknowledgements

The work presented in this dissertation would not have been possible without the help and support of a large number of people. The author first expresses his heartiest gratitude to his guide and supervisor **Dr. B.B. Biswal,** Professor, Department of Mechanical Engineering, National Institute of Technology, Rourkela for his valuable guidance, help and encouragement in the course of the present work. The successful and timely completion of the work is due to his constant inspiration and constructive criticism. The author cannot adequately express his appreciation to him. I record my deepest gratitude to Madam Biswal for family support all the times during our stay at Rourkela.

I take this opportunity to express my deepest gratitude to **Prof. K.P. Maity**, and **Prof. R.K. Sahoo,** Head, Department of Mechanical Engineering and also faculty members of the Department of Mechanical Engineering, NIT Rourkela for constant advice, useful discussions, encouragement and support in pursuing the Ph.D work.

The help and cooperation received from the **Principal, Head of the Department** of Mechanical Engineering, IGIT, Sarang are gratefully acknowledged.

The author also expresses his thankfullness to **Mr. Ratnakar Dash**, **Mr.R.N.Mahapatra** and **Mr. S.K.Panigrahy**, Researchers in NIT Rourkela for unhesitating cooperation extended during the tenure of the research programme.

Last but not the least the author expresses his thankfulness to his father, mother, wife, son, sisters and relatives for their prayer and moral support during the tenure of his work.

**Bibhuti Bhusan Choudhury**

# Abstract

Multirobot systems (MRS) hold the promise of improved performance and increased fault tolerance for large-scale problems. A robot team can accomplish a given task more quickly than a single agent by executing them concurrently. A team can also make effective use of specialists designed for a single purpose rather than requiring that a single robot be a generalist. Multirobot coordination, however, is a complex problem. An empirical study is described in the thesis that sought general guidelines for task allocation strategies. Different strategies are identified, and demonstrated in the multi-robot environment.

Robot selection is one of the critical issues in the design of robotic workcells. Robot selection for an application is generally done based on experience, intuition and at most using the kinematic considerations like workspace, manipulability, etc. This problem has become more difficult in recent years due to increasing complexity, available features, and facilities offered by different robotic products. A systematic procedure is developed for selection of robot manipulators based on their different pertinent attributes. The robot selection procedure allows rapid convergence from a very large number of candidate robots to a manageable shortlist of potentially suitable robots. Subsequently, the selection procedure proceeds to rank the alternatives in the shortlist by employing different attributes based specification methods. This is an attempt to create exhaustive procedure by identifying maximum possible number of attributes for robot manipulators.

Availability of large number of robot configurations has made the robot workcell designers think over the issue of selecting the most suitable one for a given set of operations. The process of selection of the appropriate kind of robot must consider the various attributes of the robot manipulator in conjunction with the requirement of the various operations for accomplishing the task. The present work is an attempt to develop a systematic procedure for selection of robot based on an integrated model encompassing the manipulator attributes and manipulator requirements. The

developed procedure can advantageously be used to standardize the robot selection process with view to perform a set of intended tasks. The work is also aimed at creating an exhaustive list of attributes and classifying them into different distinct categories. The different methods of robot selection on the basis of fittness, capability, task requirement and case based approach are discussed in this thesis.

One of the most important aspects in the design of MRS is the allocation of tasks among the robots in a productive and efficient manner. Optimal solutions to multirobot task allocation (MRTA) can be found through an exhaustive search. Since there are $n \times m$ ways in which m tasks can be assigned to n robots, an exhaustive search is often not possible. Task allocation methodologies must ensure that not only the global mission is achieved, but also the tasks are well distributed among the robots. This thesis presents different task allocation methodologies for MRS by considering their capability in terms of time and space.

In product assembly, optimized sequence is a prerequisite for automated systems. The assembly process can be further optimized through appropriate selection and allocation of the given tasks in a multi-device framework. These two discrete tasks need to be integrated to produce the optimum result and a cost effective system. In a MRS the possibility of parallelism need to be explored for making it time efficient. To cope with the needs of the system, the present work generates an automatic assembly sequence for multirobots and seeks for optimal allocation of tasks amongst the available robots. Task allocation methodologies must ensure that not only the global mission is achieved, but also the tasks are well distributed among the robots. An effective task allocation approach considers the capabilities of the deployable robots, and then it appropriately allocates the tasks the candidate robots.

In order to make the system more practical and user friendly, the developed methodologies have been tried with an industrial problem. An integrated approach for assembly sequence generation and task allocation for MRS has been presented by considering their time and space. A 21 part drive assembly is given to illustrate the concept and procedure of the proposed methodology.

The different approaches are adopted to optimize the allocation process. Several allocation methodologies are available in texts for task allocation under various conditions. The following methods are picked up for task assignment to the robots. These are Greedy Heuristics, Linear Programming, Mixed Integer Linear Programming, Knapsack Algorithm, Hungarian Algorithm and Particle swarm optimization. PSO has the less optimimal solutions as compared to the other methodologies. Computational results indicate that the PSO is effective and efficient in solving problems of a big size as compared to other methods and PSO achieves the global solution. The results and the subsequent recommendations for MRS of different types and sizes will be handy for the planners and users in indices.

# Contents

## 3. Problem Statement and Strategies



## 4. Selection of Robots



## 5. Task Allocation in MRS

# List of Figures

# List of Tables

# Nomenclature

## Symbol

| | |
|---|---|
| State space of a robot | *S'* |
| Time requirement of a robot | *T* |
| Environment | E |
| Bid | B |
| Task | T' |
| Action | A' |
| Decision matrix | D |
| Element of the decision matrix | $d_{ij}$ |
| Attribute in the row | $j_{th}$ |
| Robot | $i_{th}$ |
| Number of short-listed robots | m |
| Number of pertinent attributes is | n |
| Normalized specification matrix, | N |
| An element of the normalized matrix | $n_{ij}$ |
| Weight vector | w |
| The weighted normalized matrix | V |
| Separation measures for +ve benchmark robots | $S_i^*$ |
| Separation measures for -ve benchmark robots | $S_i^-$ |
| Relative closeness to the +ve benchmark robot | C* |
| Relative importance matrix | A |
| Eigen value of A | λ |
| Ranking factor | σ |
| Rectangular | RE |
| Cylinderical | CY |
| Spherical | SP |
| Articulated | AR |
| Non-servo | NS |
| Servo Point-to-Point | PTP |

| | |
|---|---|
| Servo Continious Path | CP |
| Combined PTP and CP | PTP & CP |
| Hydraulic | H |
| Electric | EL |
| Pneumatic | PN |
| Lead through teach Programming; | LT |
| Teach-pendant Programming | T |
| On-line Programming | O |
| Off-line Programming | OF |
| Task-oriented Programming | TO |
| Confirmed case | *C'* |
| Query case | $Q$ |
| Index set, I = {1, 2, ..., n} | $I$ |
| Number of cases | $i$ |
| Similarity case | S' |
| Factor index set,J={ 1,2,…,n} | $J$ |
| Weight | W |
| Number of robot types | $k$ |
| Robot type index set, K = {1, 2, ..., $k$} | K |
| Number of workstations | $n$ |
| Maximum number of type $k$ robots | $m_k$ |
| Normalized time requirement of workstation $i$ when served by a type $k$ robot | $t_{ik}$ |
| Normalized space requirement of workstation $i$ when served by a type $k$ robot | $s_{ik}$ |
| Decision variable | $x_{ik}$ |
| Initial position of an object | Y |
| Target point | Z |
| Maximum reach | R |
| Workstation's space requirement in degrees | $\theta$ |
| Load balance factor | $\Delta_k$ |

| | |
|---|---|
| Allocation cost | $\delta_{ik}$ |
| Adjusted demand | $a_{ik}$ |
| Diameter | *D'* |
| Prismatic motion needed during the Task | P' |
| Revolute motion needed during the Task | R' |
| Swept area | *S* |
| Vector of variables | x |
| Vectors of (known) coefficients | C |
| Vectors of (known) coefficients | b |
| Objective function | $C^T X$ |
| Profit of task i when selected for robot j | p' [i, j] |
| Capacity | c' |
| Bounded amount of each item type j | $m_j$ |
| Worth | $p_j$ |
| Weighs | $w_j$ |
| Space requirement of task | $S_{ik}$ |
| Average space capacity | $S_{avk}$ |
| Cost function of a robot type | $f_k$ |
| Liaisons | *L* |
| Puma 560 Robot | P |
| Adept One XL Robot | A |
| Equivalent time | $\epsilon_i$ |
| Maximum Reach | MR |
| DOF | DF |
| Payload | PL |
| Velocity | VL |
| Arm geometry | AG |
| Actuator | AT |
| Control mode | CM |
| Repeatability | RT |
| Robot programming | RP |

| | |
|---|---|
| Space | SC |
| Time | TE |
| DOF | DF1 |
| Force | FR |

# Abbreviations

| | |
|---|---|
| ASIMO | Advanced Step in Innovative Mobility |
| EMAS | Edinburg Modular Arm System |
| FDA | Food and Drug Administration |
| SSRMS | Space Station Remote Manipulator System |
| IFR | International Federation of Robotics |
| UNECE | United Nations Economic Commission for Europe |
| MRS | Multi-robot system |
| MAS | Multi agent system |
| SRS | Single robot systems |
| MADM | Multiple Attribute Decision Making |
| RCDP | Robotic cell design problem |
| GA | Genetic algorithm |
| CIM | Computer-integrated manufacturing |
| RS/WA | Robot selection and Work station Assignment |
| DEA | Data envelopment analysis |
| TOPSIS | Technique for Order Preference by Similarity to Ideal Solution |
| MCDM | Multi-criteria decision making |
| DMU | Decision making unit |
| FA/C | Functionally-Accurate Cooperative |
| BLE | Broadcast of Local Eligibility |
| PAB | Port attributed behavior |
| MRTA | Multi-robot task allocation |
| NTG | Nonlinear trajectory generation |
| AHS | Advanced Highway Systems |
| MICA | Mixed Initiative Control of Automa |
| ORS | Operations and Resources Supervisory |
| TCT | Team Composition and Tasking |
| TDT | Team Dynamics and Tactics |
| CPP | Cooperative Path Planning |

| | |
|---|---|
| VDC | Vehicle Dynamics and Control |
| FAC | Flexible Assembly Cell |
| PMX | Partially Matched Crossover |
| SA | Simulated Annealing |
| CNP | Contract Net Protocol |
| FPSB | First-price sealed-bid auction |
| DOF | Degree of Freedom |
| MTBF | Mean Time between Failure |
| MTTR | Mean Time to Repair methods |
| N.V | Normalized value |
| RSA | Robot selection and assignment |
| IP | Integer program |
| AH | Assignment Heuristic |
| AM | Allocation Model |
| GH | Greedy Heuristics |
| LP | Linear Programming |
| MILP | Mixed Integer Linear Programming |
| KA | Knapsack Algorithm |
| HA | Hungarian Algorithm |
| PSO | Particle swarm optimization |
| FFOD | First Fit by Ordered Deviation |
| MIP | Mixed-integer programming |
| MINLP | Mixed-integer nonlinear programs |
| ACO | Ant Colony Optimization |
| TSP | Traveling salesman problem |
| JIT | Just-in-time |

# Chapter-I

## INTRODUCTION

# CHAPTER 1

## Introduction

### 1.1 Background of the research work

A robot is a virtual or mechanical artificial agent. In practice, it is usually an electro-mechanical system which, by its appearance or movements, conveys a sense that it has intent or agency of its own. The word robot can refer to both physical robots and virtual software agents, but the latter are usually referred to as bots. There is no consensus on which machines qualify as robots, but there is general agreement among experts and the public that robots tend to do some or all of the following actions: move around, operate a mechanical limb, sense and manipulate environment, and exhibit intelligent behavior, especially those which mimics humans or other animals. Stories of artificial helpers and companions and attempts to create them have a long history but fully autonomous machines only appeared in the 20$^{th}$ century. The first digitally operated and programmable robot, the Unimate, was installed in 1961 to lift hot pieces of metal from a die casting machine and stack them. Today, commercial and industrial robots are in widespread use performing jobs more cheaply or with greater accuracy and reliability than humans. They are also employed for jobs which are too dirty, dangerous or dull to be suitable for humans. Robots are widely used in manufacturing, assembly and packing, transport, earth and space exploration, surgery, weaponry, laboratory research, and mass production of consumer and industrial goods.

People have a generally positive perception of the robots they actually encounter. Domestic robots for cleaning and maintenance are increasingly common in developed countries. Of late, robots have gained importance in every field of work, as they have greatly shrunken the workload that has to be done by man himself. Most robots of today have little more than a mechanical arm and a computer memory. The memory allows the arm to perform the motions, either it may be stretching or for lifting anything up. The collection of motions is stored in the memory, which easily enables the robots to switch from one motion into another form in quick time.

Robots in antiquity and through the Middle Ages were used primarily for entertainment. However, the 20$^{th}$ century featured a boom in the development of industrial robots. Through the rest of the century, robots changed the structure of society and allowed for safer conditions for labor. In addition, the implementation of advanced robotics in the military and NASA has changed the landscape of national defense and space exploration. Robots have also been influential in the media and profitable for toy manufacturers.

### 1.2 Multi-robot systems

A multi-robot system (MRS) is one of the methodologies to give certain ability to a robot system. This approach expects emerging of new abilities through just simple and small interactions among multiple robots. The new abilities are not expected in a single robot system. An emerged ability is expected in flexibility, adaptability robustness. Although many researches who are interested in these ideas have investigated, almost all the research mainly focuses on locomotion, formation or reconfiguration of MRS. To expand probabilities and expectationsfor emergent robotics, on the other hand, focus on sensing by MRS. Sensing situations of robot systems will be needed in its adaptive behavior, which is also including such locomotion, formation and reconfiguration. There are two interactions in a MRS. One is physical interaction and the other is informational one.

The time required to reach other planets makes planetary surface exploration missions prime targets for automation. Sending rovers to other iplanets instead of or together with people can also significantly reduce the danger cost involved. Teams of rovers are both more fault tolerant (through redundacy) and more efficient (through parallelism) than single rovers if the rovers coordinated well. However, rovers cannot be easily tele-operated since this requires a large number of human operators and is communication intensive, error prone, and slow. Neither can they be fully preprogrammed since their activities depend on their discoveries. Thus, one needs to endowthem with the capability to coordinate autonomously with each other. It should be pointed out that the important applications of robots are by no means limited to those industrial jobs where the robot is directly replacing a human worker. There are many other applications of robotics in areas where the use of human is impractical or undesirable. Among these are under-sea and planetary exploration, satellite retrieval and repair, the defusing of explosive devices, and work in radioactive environments.

Multiple cooperating robots hold the promise of improved performance and increased fault tolerance for large-scale problems. For many applications, a team of robots can be effectively used and it can accomplish a given task more quickly than a single agent can by dividing the task into sub-tasks and executing them concurrently. A team can also make effective use of specialists designed for a single purpose (e.g., scouting an area, picking up objects, hauling payload), rather than requiring that a single robot be a generalist, capable of performing all tasks but expert at no tasks. A group of collaborating robots performs certain tasks better than a single robot. For many applications using more than one robot to perform a specific task has many potential advantages over a single robot configuration. In short, a population of cooperative robots behaves like a distributed robot to accomplish tasks that would be difficult, if not impossible, for a single robot. However, the advantages of MRS are often offset by the complexity in achieving a successful implementation. The complex problem of multi-robot coordination can be considered in the framework of multi-robot dynamic task allocation. This problem can be considered in the framework of multi-robot dynamic task allocation under uncertainty.

Research performed under such titles as distributed robotic systems, swarm robotics, decentralized robotic and multi-agent robotics has focused on the investigation of issues and applications of systems composed of groups of robots. The general idea is that teams of robots, deployed to achieve a common goal, are not only able to perform tasks that a single robot is unable to, but also can outperform systems of individual robot, in terms of efficiency and quality. In addition, groups of robots provide a level of robustness, fault tolerance, and flexibility, as the failure of one robot does not result in the unsuccessfulness of the mission, as long as the remaining robots share the tasks of the failed robot. Examples of tasks appropriate for robot teams are large area surveillance, environmental monitoring, large object transportation, planetary exploration, and hazardous waste cleanup.

Applications of robot teams are in four basic areas, where the requirement may be as follows;

i. Large objects must be handled
ii. Large areas must be covered
iii. Iterative tasks must be performed and
iv. Robustness and fault tolerance is required.

There are a number of certain situations that lends themselves well to the task decomposition and allocation among multiple robots. The most significant concept in MRS is cooperation. It is only through cooperative task performance that the superiority of robot groups can be demonstrated. The cooperation of robots in a group can be classified into two categories of implicit cooperation and explicit cooperation. In the implicit cooperation case each robot performs individual tasks, while the collection of these tasks is toward a unified mission. For example, when multiple robots are engaged in collecting rock samples and returning them to a common place, the team is accomplishing a global mission while cooperating implicitly. This type of group behavior is also called asynchronous cooperation, as it requires no synchronization in time or space. The explicit cooperation is the case where robots in a team work synchronously with respect to time or space in order to achieve a goal. One example of such cooperation is transportation of heavy objects by multiple robots, each having to contribute to the lifting and moving of the object. This task requires the robots to be positioned suitably with respect to each other and to function simultaneously. Regardless of the type of cooperation, the goal of the team must be transformed into tasks to be allocated to the individual robots. Multi-robot teamwork is a complex problem consisting of task division, task allocation, coordination, and communication.

MRS have been proposed in the last decade in a variety of settings and frameworks, pursuing different research goals, and successfully applied in many application domains. Special attention has been given to MRS developed to operate in dynamic environments, where uncertainty and unforeseen changes can happen due to the presence of robots and other agents that are external to the MRS itself. Generally speaking, an MRS can be characterized as a set of robots operating in the same environment. However, robotic systems may range from simple sensors, acquiring and processing data, to complex human-like machines, able to interact with the environment in fairly complex ways. Moreover, it is not easy to give a definition of the level of autonomy that is required for a robot in order to be considered an entity acting in the environment, as opposed to a simple machine that provides services to the operator (a printer or a even a light switch). The subset of MRS can be further characterized as the one that is addressed by considering three main aspects: (i) the rationale for the design of the MRS, (ii) the basic functionalities and technologies (both hardware and software) used in the MRS development and (iii) the tasks that the robots should perform and the intended application domains. From an engineering stand point, the MRS can improve the effectiveness of a robotic system either from the viewpoint of the performance in accomplishing certain tasks, or in the robustness and reliability of the system, which can be increased by modularization. The coordination of candidate robots in MRS with respect to the system can be suitably planned from the view point of strategic implementation. The coordination dimension can be related to the system dimension as presented in Table 1.1. The MRS taxonomy is mentioned in Figure 1.1.

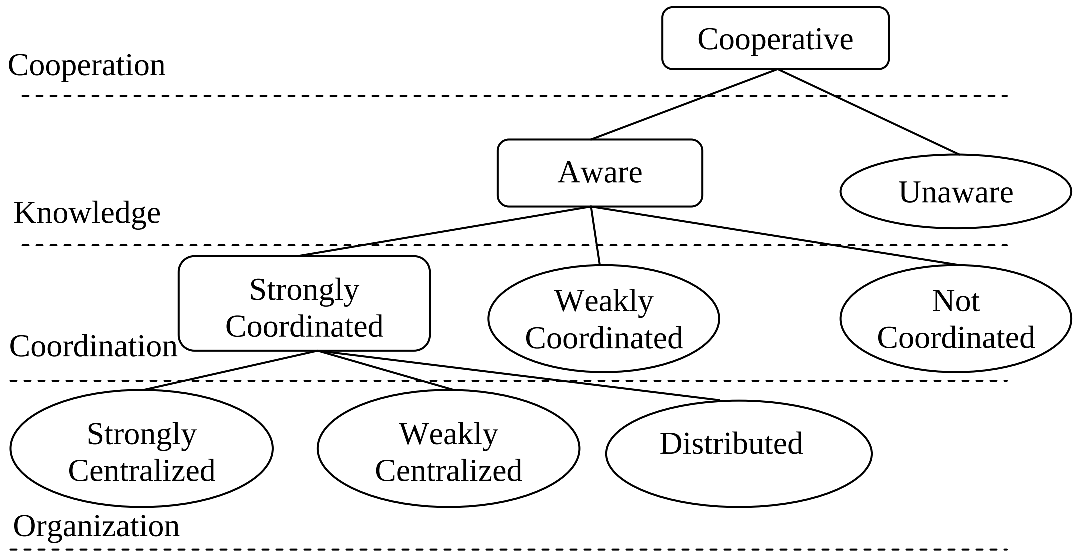



Figure 1.1 MRS taxonomy

Table 1.1 Classification and dimensions

| Coordination Dimensions | System Dimensions |
|---|---|
| Cooperation | Communication |
| Knowledge | Team Composition |
| Coordination | System Architecture |
| Organization | Team Size |

*Cooperation Level:* The first level is concerned with the ability of the system to cooperate in order to accomplish a specific task. At this level cooperative systems are distinguished from not cooperative ones. A cooperative system is composed of robots that operate together to perform some global task.

*Knowledge Level:* The second level of the hierarchical structure is concerned with the knowledge that each robot in the team has about its team mates. Aware robots have some kind of knowledge of their team mates, while unaware robots act without any knowledge of the other robots in the system. The interest in cooperating unaware MRS is motivated from an engineering point of view by the simplicity of such systems, with respect to aware ones.

*Coordination Level:* The third level is concerned with the mechanisms used for cooperation in which the actions performed by each robotic agent take into account the actions executed by the other robotic agents in such a way that the whole ends up being a coherent and highperformance operation. However, there are different ways a robot can take into account the actions of the other members of the team. The underlying feature is the coordination protocol that is defined as a set of rules that the robots must follow in order to interact with each other in the environment.

*Organization Level:* The fourth level of our hierarchical structure is concerned with the way the decision system is realized within the MRS. This level introduces a distinction in the forms of coordination, distinguishing centralized approaches from distributed ones. In particular, a centralized system has an agent (leader) that is in charge of organizing the work of the other agents; the leader is involved in the decision process for the whole team, while the other members can act only according to the directions of the leader. On the other hand, a distributed system is composed of agents which are completely autonomous in the decision process with respect to each other; in this class of systems a leader does not exist. The classification of centralized systems can be further refined depending on the way the leadership of the group is played. Specifically, strong centralization is used to characterize a system in which decisions are taken by the same pre-defined leader agent during the entire mission duration, while in a weakly centralized system more than one agent is allowed to take the role of the leader during the mission. Along with the classification introduced to characterize the form of coordination, there are a number of system features that are relevant to the development of the system. They can be grouped in the system dimensions, which include: communication, team composition, system architecture and team size.

*Communication:* Cooperation among robots is often obtained by a communication mechanism that allows the robots to exchange messages. A detailed analysis of the various technical problems related to communication in Multi agent system (MAS) is given for example in [1]. However, when MRS are considered the communication mechanisms are very different; in addition most of the MRS that operate with a limited number of robots (i.e. less than 10), except for a few recent projects for large-scale MRS that take into account about 100 robots, while in large-scale MAS the number of agents can often be in the order of 10,000-100,000. These observations show that communication issues have, in general, different characteristics for MAS and MRS. There can be two different types of communication depending on the way the robots exchange information: direct or indirect communication. Direct communication makes use of some on board dedicated hardware device, while indirect communication makes use of stigmergy [2]. The fact that in MRS direct communication is based on a dedicated physical devices, results in a much more expensive and unreliable solution to attain coordination with respect to MAS. Therefore, indirect communication has received particular attention in MRS literature, to cut implementation and design costs. Stigmergic communication can both guarantee locality in the interactions among agents, reducing the complexity for the design of large scale systems, and avoid the need of synchronization between the agents, by providing a shared communication structure that each agent can access in a distributed concurrent fashion.

*Team Composition:* According to team composition MRS can be divided in two main classes, heterogeneous and homogeneous. Homogeneous teams are composed of team members that have exactly the same hardware and control software, while in heterogeneous teams the robots differ either in the hardware devices or in the software control procedures. This distinction is used also for MAS, but in that case the differences are obviously only in the software implementation of the agents' behaviors.

*System Architecture:* System architecture is an important feature for classifying MRS as well as MAS. The architecture refers to the whole MRS and not to the architecture of the single robotic agent. A precise characterization of MRS with respect to reactive or deliberative architectures is presented in [3]. Team architecture is considered as deliberative if it allows the team members to cope with the environmental changes by providing a strategy to reorganize the overall team behaviors. On the other hand, in reactive team architectures each robot in the team copes with the environmental changes by pursuing an individual approach to reorganize its own task in order to accomplish the goal assigned to it. The main difference between deliberative and reactive team architectures relies on the different approaches adopted by the MRS to recover from an unpredicted situation: in a deliberative MRS a long term plan involving the usage of all the available resources to collectively accomplish a global goal is provided; in a reactive MRS a plan to cope with the problem at hand is provided by the robotic agent directly involved with it.

*Team Size*: The team size is an important issue for MAS and it is becoming a relevant issue also in MRS development, actually a number of recent works explicitly address large scale MRS [4, 5]. However, the number of robots acting in the same environment is still quite limited with respect to the number of agents in MAS.

Some of the multiple issues that can be addressed by the proper task allocation mechanism are:

- The reason for robots to function in a group.
- Whether all robots have a unique goal like soccer team or they have a multiple goals such as a free market system.
- Whether the robots act in a self-centered manner or as team-aware individuals.
- The mutual cooperation amongst the robots under focus.

### 1.2.1 Classification of MRS

There are many types of MRS each capable of performing a wide variety of tasks. Due to the wide variety of devices and configurations that may be classed as multi-robot, some form of classification is required to put these systems into perspective. The classification robot systems as presented in Figure 1.2 give an indication of the broad scope of MRS. The robots are first classified into fixed base or mobile categories, with the fixed base category being subdivided into two components termed independent and coordinated, which may also be referred to as loosely and tightly coupled systems respectively. The independent fixed base systems comprise of a set of fixed base robots working within a common workspace but performing independent tasks. These tasks are usually subtasks of the global task for the workcell, for example using multi-robots to perform simple pick and place operations off a conveyor belt, providing a higher throughput than could be achieved by using a single robot device.

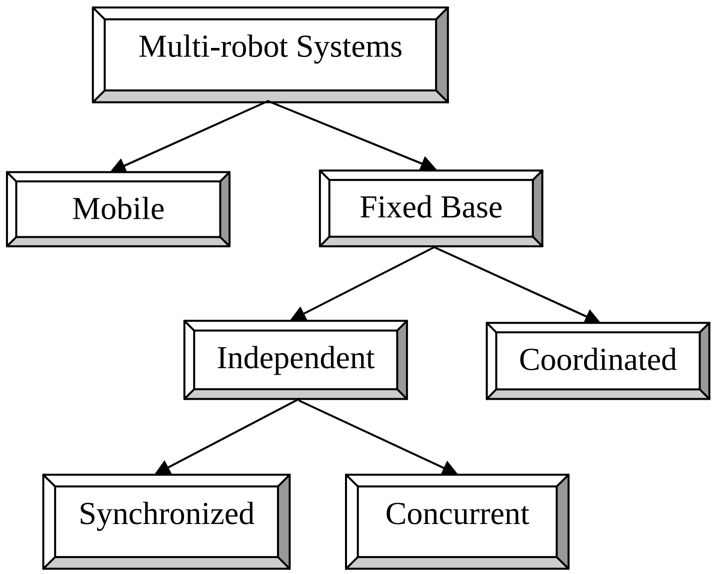



Figure 1.2 MRS classifications

This class of system may be subdivided further into synchronized and concurrent systems. Synchronized systems are configured such that at each time instant only one robot may be working in the common workspace between the groups of robot, the common workspace being an exclusively shared resource. Concurrent systems, on the other hand, are more sophisticated and enable more than one robot to operate in the common workspace simultaneously. Coordinated fixed based systems comprise of a set of fixed base robots performing the same task concurrently, for example two robots handling a heavy object such as a beam. In this situation the robots act as a closed kinematic loop.

### 1.2.2 Types of MRS

*A. Homogeneous and heterogeneous systems*

One main issue in task allocation is the division of the tasks into homogeneous versus heterogeneous tasks. Their implementation may range from homogenous system where all robots have the same task to a grouping, which divides the robots in different groups, and each group is assigned to do a different task. They may use inference and temporal parameters to evaluate different methods. The best performance is obtained through homogenous task allocation, i.e., the fastest collection of trash than others. It is too difficult to build a team of large number of robots, make sure that all are functioning and perform experiments with them. Instead, the researchers have been conducting the hardware experiments with only a few robots, and then they have augmented their hardware studies with computer modeling and simulation of robot groups with large populations. It should be noted that the effects of team size and its scaling are integral issues in robot group studies, and the reliability of the simulation results remains to be seen. In some simulation and analytical studies, the focus is on complex emergent behavior of a collection of simple robots, i.e., collective behavior. These works use mathematics to predict and design working group of robots.

### 1.2.3 Different options for allocation in MRS

This way, robots can develop special relations with specific other robots. These relations are:

i) Single robot performing single task
ii) Single robot performing multiple tasks
iii) Multi-robot (homogeneous) performing single tasks
iv) Multi-robot (heterogeneous) performing single task
v) Multi-robot (homogeneous) performing multiple tasks
vi) Multi-robot (heterogeneous) performing multiple tasks

### 1.2.4 Single robot vs multi-robot

An MRS is composed of multiple, interacting robots. The study of MRS has received increased attention in recent years. This is not surprising, as continually improving robustness, availability, and cost-effectiveness of robotics technology has made the deployment of MRS consisting of increasingly larger numbers of robots possible. With the growing interest in MRS comes the expectation that, at least in some important respects, multiple robots will be superior to a single robot in achieving a given task. The benefits of a MRS over a SRS (Single robot systems) are outlined in order to introduce issues involved in MRS control and study their similarity and differences. The study of MRS has received increased attention in the recent years. This is not surprising as continually improving technology has made the deployment of MRS consisting of increasingly larger number of robots possible. It is obvious that, at least in some important respects, multiple robots will be superior to a single robot in achieving a given task. Potential advantages of MRS over a SRS include reduction of total system cost by employing multiple simple and cheap robots as opposed to a single, complex and expensive robots. Furthermore, the inherent complexity of certain task environment may require the use of multiple robots as the demand for capability is quite substantial to be met by a single robot. Multiple robots are assumed to increase system robustness by taking advantage of inherent parallelism and redundancy. Multi-robot teamwork is a complex problem consisting of task division, task allocation, coordination, and communication. One of the significant concepts in MRS is cooperation. It is only through cooperative task performance that the superiority of robot groups can be demonstrated. The cooperation of robots in a group can be classified into two categories of implicit cooperation and explicit cooperation. In the implicit cooperation case each robot performs individual tasks, while the collection of these tasks is toward a unified mission. This type of group behavior is also called asynchronous cooperation, as it requires no synchronization in time or space. The explicit cooperation is the case where robots in a team work synchronously with respect to time or space in order to achieve a goal. One example of such cooperation is transportation of heavy objects by multiple robots, each having to contribute to the lifting and moving of the object.

This task requires the robots to be positioned suitably with respect to each other and to function simultaneously. Regardless of the type of cooperation, the goal of the team must be transformed in to tasks to be allocated to the individual robots.

Distributed MRS stand in contrast to centralized MRS, in which each robot's actions are not completely determined locally, as they may be determined by an outside entity, such as another robot or by any type of external command. In distributed MRS, each robot must make its own control decisions based only on limited, local, and noisy sensor information. The consideration is limited to distributed MRS because they are the most appropriate for study with regard to systems that are scalable and capable of performing in uncertain and unstructured realworld environments where uncertainties are inherent in the sensing and action of each robot. Strictly speaking, the issues in a centralized MRS are more akin to a scheduling or optimal assignment and less of a problem of coordination in a distributed system.

## 1.3 Robot performance and selection

A robot is characterized by its degree of freedom, number of joints, type of joints, joint placement, link lengths and shapes, and their orientation which influence its performances. The speed of operation significantly depends on the complexities of the kinematic and dynamic equations and their computations. Aspects of kinematics and dynamics should be looked into for selecting a suitable robot. Usually, kinematic characteristics like workspace, etc. are considered for the selection of a robot for an application. Robots with vastly different capabilities and specifications are available for a wide range of applications. Various considerations such as availability, management policies, production systems compatibility, and economics need to be considered for selecting a suitable robot. The complexity of problem can be better appreciated when one realizes that there are large numbers of attributes that have to be considered in the selection of robot for particular application. Moreover, many of them are conflicting in nature and have different units, which cannot be unified and compared as they are. However, none of these solutions may take care of all the demands and constraints of a specific application. There are a number of reported studies concerning the selection of robots for manufacturing applications.

Selecting the right kind of robot for an application is not easy. In addition, just meeting the customer requirements can be a challenge. The addition of system integration in workcell design processes may further complicate the picture. In the robot market today, there are many robot manufacturers with number of robot configurations. There has been rapid increase in the number of robot systems and robot manufacturers. Fortunately, a number of tools and resources are becoming available to help designers select the most suitable robot for a new application. However, none of these solutions can take care of all the demands and constraints of a user specific robotic workcell design. Eventually the designers must use the available information and make their own decisions.

## 1.4 Objective

The initial study of some relevant literatures in the area of MRS for industrial application clearly points towards some general issues. These issues are identified as

i) Selection of robots for MRS.

ii) Strategies for employing the robots and that for coordinating/controlling them under MRS.

iii) Allocation of tasks to the robots with a view to conveniently handle the desired tasks, to utilize the robots under question to the maximum extent possible, and to minimize the throughput time.

Under this backdrop, the objectives of the present research work are outlined as follows.

- To make an extensive study on the subject concerned as well as the research activities already carried out in the area and thereafter enumerating and analyzing the pros and cons of various methodologies. This will make the understanding of the problem better about the real areas of concern and appreciating the scope for improvement

- To study the specific area of mathematical theories having greater capabilities for application in MRS and its issues under various conditions.
- To develop a correct and scientific method for selection of robots for MRS.
- To find out an appropriate task allocation methodology for industrial application under multi-robot environment.
- To maximize the utilization of the candidate robots in MRS, using appropriate methodology of task allocation.

Apart from these broad objectives the present research work also addresses several related issues such as;

- To identify the robot selection attributes, and obtain the most appropriate combination of the attributes in conjunction with the real requirements of the industrial application.
- To conduct an empirical study that seeks general guidelines for task allocation strategies in systems of multiple cooperating robots. Task allocation strategies need to be identified that aim at studying tradeoffs between commitment and coordination.
- To develop an integrated approach for assembly sequence generation and task allocation for MRS by considering their capability in terms of time and space

## 1.5 Scope of the work

The domain of robotic application in industry, the environmental conditions, the dynamism, the strategies etc. while looking at general MRS can be very large. The present work is envisaged under its own scope of study. The various methodologies for task allocation under various conditions are studied, which are suitable to industrial robots with the listed options. The research work is restricted to the implicit cooperation where robot performs the individual tasks, while the group of these tasks is toward a unified assignment. It is assumed that each robot is competent of estimating its robustness for every task it can execute. The factors such as economic considerations, availability, management constraints and corporate policies, international market policies etc. for the selection of robots are beyond the scope of this work. In the present work as many as 30 attributes of the robots are identified and an attempt has been made to codify most of the robot characteristics, which will define the robot precisely and accurately. Future research will involve both improvements in solution methods and extensions to the current model.

### 1.6 Organization of the thesis

This thesis is divided into seven chapters, described as follows:

Chapter 1 provides an overview and introduction of the research. Research background, motivation, aims and objectives are elaborately described. Chapter 2 reviews on several diverse streams of literature on different issues of the topic such as strategies, selection, task allocation, task assembly optimization techniques etc. In consequence, the research gaps are identified. Chapter 3 discusses the problem statement and strategies of robots for allocation. In this chapter attempts are made to empirically derive some guidelines for selecting task allocation strategies for MRS with implicit cooperation. Chapter 4 presents different methods suitable for selection of candidate robots for the problems under consideration and then the methods and procedures are detailed. Chapter 5 presents the general framework that is used in the thesis to model problem-solving in an MRS, and uses theoretical examples to illustrate the different task allocation strategies in systems of multiple cooperating robots. This chapter also introduces the different optimization techniques that are used for the work to solve various MRS problems. The various optimization algorithms for achieving better results are stated. The solution methodologies of these techniques are presented through coding them in Lingo, Matlab, and Management Scientist as applicable. Chapter 6 discusses the outcome of the research and also identifies the pros and cons of different methods. A comparative study of these methods is made in the light of the strategies, selection and task allocation in MRS.  The conclusions on different aspects of the entire work are presented in Chapter 7 along with the directions for future work.

## 1.7 Summary

There is a growing demand for teams of multiple robots to be employed in many application domains. Multi-robot solutions are especially desired for tasks which are too dangerous, expensive, or difficult for humans to perform. It is obvious that multiple robots achieve both more robust and more effective behavior by accomplishing coordinated tasks that are not possible for single robots. Groups of homogeneous and heterogeneous robots have a great potential for application in complex domains that may require the intelligent use and merge of diverse capabilities. The chapter presents a brief study of the subject and describes the importance of robotic applications in industries along with the areas that need focus for research and improvement.

# Chapter-II

# LITERATURE SURVEY

# CHAPTER 2

## Literature Survey

### 2.1 Introduction

As research progresses in robotic systems, more and more aspects of MRS are being explored. Several researchers began investigating issues in multiple mobile robot systems. Prior to this time, research had concentrated on either single robot systems or distributed problem-solving systems that did not involve robotic components. Since this early research in robotics, the field has grown dramatically, with a much wider variety of topics being addressed. Several new robotic application areas, such as underwater and space exploration, hazardous environments, service robotics in both public and private domains, the entertainment field, and so forth, can benefit from the use of MRS. In these challenging application domains, MRS can often deal with tasks that are difficult, if not impossible, to be accomplished by an individual robot. A team of robots may provide redundancy and contribute cooperatively to solve the assigned task, or they may perform the assigned task in a more reliable, faster, or cheaper way beyond what is possible with single robots. Some areas have been explored more extensively, however, and the community is beginning to understand how to develop and control certain aspects of multi-robot teams. Many of the research papers address more than one of these foundational problems in MRS. Therefore aspects of this work as they apply to each of these key research areas are described. For context, other key references and examples of prior research in each of these principle topic areas are also discussed. However, space does not allow an exhaustive treatment of each of these important research areas, and thus it is not possible to thoroughly review all the previous literature pertinent to this subject.

## 2.2 Scenarios

In MRS the following pertinent issues are very important to be considered:

- There is a task/problem to be solved by the system of robots.
- Robots are able to reason about what they are doing.
- Robots are allowed to communicate with each other and with a human.
- The human provides the initial goal and specifications.
- Robots are allowed to sense their environment dynamically.
- Robots carry out actions and contribute to the overall task (the mission).
- There are real-time issues that need to be addressed, such as a deadline for mission or subtask completion,
- The environment the robots are working in can change unexpectedly.

A potential robot user is now faced with many options. The decision on which robot to select is made more complex because robot performance is specified by many parameters for which there are, as yet, no industry-wide standards. Apart from this, one is faced with a challenge to wisely select robots amongst the available ones for employment in a particular application environment. The allocation of the desired tasks the coordination of the robots in MRS, the cooperation amongst the robots in action pose several issues in designing and implemanting MRS for industrial applications.

The following paragraphs present some of the major and relevant work in the area of MRS, task assignment, assignment techniques and optimization of team/group formation for creating the multirobotic work cell. Some of the important research papers with relevance to the present work are presented in Table 2.1.

Table 2.1 A summary of robot selection models

| Author | Application | Solution approach | Consideration | Selection criterion |
|---|---|---|---|---|
| E. Ertugrul Karsak | Facility site selection system | DEA And fuzzy robot selection algorithm | cost and technical performance parameters | Best combination of cost and erformance parameters. |
| M. J. Khouja and R. L. Kumar | General | Opions model | Speed,load ,repetability and price | net present value |
| Marcello braglia and Roberto Gabbrielli | General | Dimensional Analysis theory | Velocity,Load capacity, Cost, repetability, Vendors' service, Programming flexibility | As per ranking |
| R. Venkata Raoa, K.K. Padmanabhan | General | digraph and matrix methods | Purchase cost, load capacity, velocity, repeatability, DOF and man-machine interface. | As per the value of robot selection index |
| Moutaz Khouja, David E. Booth, Michael Suh and John K. Haney Jr | Robotic assembly cells | fuzzy cluste ring algorithm | Reach,load, repetability | As per grade of membership |
| Agrawal et el. | General | Multiple Attribute Decision Making | Engineering Attributes | DM's utility |
| Booth et el. | General | Statistical | Engineering Attributes | Maximum Mahalanobis distance |
| S.C. Botelhoand R. Alami | Hospital environment | M+ task achievement | re-scheduling, suppression of redundancies and opportunistic enhancement | As per simulation results |
| C. Micacchi and R. Cohen | RoboCup Search and Rescue, | Simulation Modelling | Unexpected events | As per the simulation scenario |
| Lynne E. Parker | Hazardous waste cleanup | ALLIANCE | fault tolerant, reliable, and adaptive | As per the ALLIANCE architecture |
| Brian P. Gerkey and Maja J MatariC | Cooperatively reallocate a large box to a specified goal | MURDOCH | pusher-watcher | Using the Auction based |

| | | | | |
|---|---|---|---|---|
| M. Berhauld ,H. Huang P. Keskinocaki, S. Koenigi, W. Elmaghrabyi, P. Griffin, A. Kleywegd | General | Combinatorial Auctions | combinatorial bidding strategies | As per the Graph-cut |
| Robert Zlot and Anthony Stentz | Reconnaissance scenario | complex task allocation problem | novel task tree auctions | As per task tree allocation mechanism |
| A.Sahu and R. Tapadar | General | Genetic Algorithm and Simulated Annealing | Partially Matched Crossover (PMX) And exponential cooling schedule based on Newtonian cooling | As per the Optimized one |

### 2.3 Models for selection of robots

Research on the industrial robot selection problem has received increased attention in the past decade. In this chapter, the models are reviewed. The strengths and weaknesses of the different approaches to the robot selection problem are summarized. A tabular framework is used to summarize the reviewed models. For quick and easy reference, the table categorizes the models by application, solution approach, robot attributes considered, and selection criteria.

Vukobratovic [6] found that the spherical configuration was superior to the jointed-arm, cylindrical, or rectangular robot designs in terms of speed and energy consumption.Robot selection problem has received increasing attention in the past decade, parallel to the upward trend in the usage of industrial robots. A number of researchers have developed computer-aided procedures to address the robot selection problem. Agrawal et al. [7] used Multiple Attribute Decision Making (MADM) approach to generate a completed “query problem” for robot selection. Khouja et.al. [8] consider the problem of selecting robots for an assembly cell which produces several products, each of which requires a number of tasks. Each task requires some minimal level of robot performance on attributes such as load capacity, repeatability and reach. Due to the large number of available robots and their wide range of performance, the problem of selecting robots for the cell and assigning tasks to these robots can be complex. This problem will be referred to as the robotic cell design problem (RCDP).The proposed approach recognizes and exploits the flexibility of robots. It also recognizes that the manufacturer specifications of robots do not hold simultaneously under normal operating con ditions. A numerical example is presented and a small experiment is conducted to test the procedures. Booth, Khouja, and Hu [9] used robustified Mahalanobis distance and principal components analysis to identify better performing robots. Mahalanobis distance is used to identify outlying robots while principal components analysis is used to indicate if a robot is an outlier because it provides better or worse combination of specifications from the average robot. In robustified Mahalanobis distance, the vector of means for robot attributes as specified by the manufacturers is computed. To identify outlying robots, a weight function that assigns each observation a weight that is inversely proportional to its distance from the center of the data is used iteratively to recompute the vector of means and distances until convergence occurs. At convergence, Mahalanobis distances are used to identify outlying robots. Dooner [10] simulated robot operation in the workspace and used the workspace as an aid to robot selection. Huang and Ghandforoush [11] stated a procedure to evaluate and select the robot depending on the investment, budget requirements and comparing the suppliers of the robots. But they had assumed that the user knows which robot to buy and the question was from whom to buy. Madhuraj [12] selected robot considering cost as one criterion and used to shortlist the robots for the particular applications. In the contemporary work, Hinson [13] stated that the working environment of the robot is a major selection factor. He considered work envelop of the robot for the evaluation. Jones [14] used marginal value function to evaluate and rank the robots. A number of studies are reported on the selection of robots for various applications. Paul and Nof [15] compared humans to robots in order to determine which of the two was better suited for a given job. Hinson [16] stated that the working environment of the robot is a major selection factor. He also considered the work envelop of the robot for its evaluation. A body of lite rature on the design of robot assembly cells has been developed over the past decade [17].

Offodile et al. [18] developed a coding and classification system which was used to store robot characteristics in a database, and then selected a robot using economic modeling. Liang and Wang [19] proposed a robot selection algorithm by combining the concepts of fuzzy set theory and hierarchical structure analysis. The algorithm was used to aggregate decision makers' fuzzy assessments about robot selection attributes weightings, and to obtain fuzzy suitability indices.

Rao and Padmanabhan [20] proposed a methodology based on digraph and matrix methods for evaluation of alternative industrial robots. A robot selection index was proposed that evaluates and ranks robots for a given industrial application. The index was obtained from a robot selection attributes function, in turn obtained from the robot selection attributes digraph. The digraph was developed based on robot selection attributes and their relative importance for the application considered. A step by step procedure for evaluation of a robot selection index was suggested.

Zhao and Yashuhiro [21] introduced a genetic algorithm (GA) for an optimal selection and work station assignment problem for a computer-integrated manufacturing (CIM) system. In CIM systems, Robot Selection and Work station Assignment (RS/WA) problem is very important and has significant impact to deliver high quality and low cost products on timely basis. Specifically, the RS/WA problem for a CIM system seeks the optimal combination of robots of different types to serve all given work stations such that each work station's resource demands are satisfied, no robot capacity constraint is violated, and the total system cost is minimized. Since the problem can be considered as a generalized two-dimensional multi-type bin packing, a well-known NP-hard problem, it is not possible that directly solve the problem and provide exact solutions within a reasonable time limit.

Boubekri et al. [22] developed an expert system for industrial robot selection considering functional, organizational and economical factors in the selection process. The use of data envelopment analysis (DEA) for robot selection has been addressed by Khouja [23]. Bhangale et al. [24] listed a large number of robot selection attributes, and ranked the robots using Technique for Order Preference by

Similarity to Ideal Solution (TOPSIS) and graphical methods, comparing the rankings given by these methods. However, the weights assigned by the authors to the attributes were not consistent. Karsak [25] proposed a two-phase methodology is proposed for robot selection. In phase 1, data envelopment analysis is used as a means to determine the technically efficient robot alternatives, considering cost and technical performance parameters. Using data envelopment analysis permits us to consider the fact that the performance parameters specified by the vendors are generally unattainable in practice. In the second phase, a fuzzy robot selection algorithm is utilized to rank the technically efficient robots according to both predetermined objective criteria and additional vendor-related subjective criteria. The algorithm is based on calculating fuzzy suitability indices for the technically efficient robot alternatives, and then, ranking the fuzzy indices to select the best robot alternative. Karsak and Ahiska [26] introduced a practical common weight multi-criteria decision making (MCDM) methodology using the DEA method with an improved discriminating power for technology selection. The results indicate that the proposed framework enables further ranking of DEA-efficient decision making unit (DMU) with a notable saving in the number of mathematical programming models solved.

## 2.4 Task allocation

Multi-robot teamwork is a complex problem consisting of task division, task allocation, coordination, and communication. The most significant concept in MRS is cooperation. It is only through cooperative task performance that the superiority of robot groups can be demonstrated. The cooperation of robots in a group can be classified into two categories of implicit cooperation and explicit cooperation. In the implicit cooperation case each robot performs individual tasks, while the collection of these tasks is toward a unified mission.

This type of group behavior is also called asynchronous cooperation, as it requires no synchronization in time or space simultaneously. Regardless of the type of cooperation, the goal of the team must be transformed in to tasks to be allocated to the individual robots. The explicit cooperation is the case where robots in a team work synchronously with respect to time or space in order to achieve a goal. One example of such cooperation is transportation of heavy objects by multiple robots, each having to contribute to the lifting and moving of the object. This task requires the robots to be positioned suitably with respect to each other and to function.

Teams of robotic systems at first glance might appear to be more trouble than they are worth. There are several reasons why two robots or more can be better than one
a) Distributed action: Many robots can be in many places at the same time;
b) Inherent Parallelism: Many robots can do many, perhaps deferent things at the same time;
c) Divide and conquer certain problems are well suited for decomposition and allocation among many robots; and
d) Simpler is better: Often each agent in a team of robots can be simpler than a more comprehensive single robot solution.

No doubts there are more reasons as well. Unfortunately there are also drawbacks in particular regarding coordination and elimination of interference. The degree of difficulty imposed depends heavily upon the task and the communication and control strategies chosen [27].

In many cases several mobile robots can be used together to accomplish tasks that would be either more difficult or impossible for a robot acting alone. Although most mobile robotic systems involve a single robot operating in an environment, a number of researchers have considered the problems and potential advantages involved in having an environment inhabited by a number of robots. For some specific robotic tasks, such as exploring an unknown planet, it has been suggested that rather than sending one very complex robot to perform the task it would more effective to send a large number of smaller, simpler robots. Such a collection of robots is sometimes described as a swarm. Using multiple robots rather than a single robot can have several advantages and leads to a variety of design tradeoffs. In particular, large numbers of simple robots may be simpler in terms of individual physical design and thus the ensuing system can be more economical, more scalable and less sensitive to overall failure. Likewise, destruction of a single member of a large swarm may not be catastrophic while the failure of a single subsystem of a conventional robot is usually disastrous [28].

The system per say consists of set of either homogeneous or heterogeneous roots. While looking at the application of MRS, it involves a lot many other functions. Some of the functions are task allocation, robot selection for carrying out the desired tasks, forming of the task force amongst the available robots, control, coordination and scheduling, workcell design etc. The following sections present some of the important work carried out by various researchers and agencies towards the development and growth of MRS.

The various issues and methodologies related to task allocating have been in the research and application domain since long. The applicability of this research ranges from Networkship, Multitasking shop floors, skilled personnels and so on. In the present research work, our discussion is limited to multi-robot environments. Goldberg and Mataric [29] studied homogeneous and heterogeneous task allocation for a foraging task such as trash collection. Their implementation ranged from homogenous system where all robots have the same task to a grouping, which divides the robots in different groups, and each group is assigned to do a different task. Inference, spatial, and temporal parameters are used to evaluate different methods. Their experimental result shows that the grouping system is suitable for reducing interference. However the best performance is obtained through homogenous task allocation. In a similar work Sukthanker and Sycara [30] showed that when systems that are substantially more efficient augmented by homogenous task allocation by making robots more team-aware.

The study of MRS can be dealt with in hardware with small population sizes, versus the study of issues in MAS in simulation with large population sizes. Construction, maintenance, and utilization of large groups of robots are infeasible due to time and budget constraints. This led the researchers to conducting the hardware experiments with only a few robots, and then augmenting their hardware studies with computer modeling and simulation of robot groups with large populations. The effect of team size, scaling, and the reliability of simulation are to be kept in mind while conducting studies on robot groups. Lerman et al. [31] evolved a mathematical methodology based on viewing large colonies of robots (swarms) as stochastic systems, Markov property, for predicting their emergent behavior. This analysis can be useful in many applications, as the Markov property holds good in many MRS. Obviously, mathematical analysis helps in predicting the collective emergent behavior and understands if the effects of missions are more suitable.

In geometric formation, a team of mobile robots attempts to achieve and maintain a geometrical shape while performing the given task. This type of problem has been studied by researchers [32, 33]. The static task allocation usually works well if formation is treated like a coordination problem Balch and Arkin [27, 32] and Gerkey et.al [34] proposed a method of team formation where the task allocation takes place during system design. The common approach in all these work is that all of the robots have a predefined and similar task. This work essentially used a schema-based architecture [35] to implement motor schema navigation. The schemas are activated in parallel by percenting the second data. These asynchronous processes start behaviors generated in vector format is multiplied by an importance weight. The sum of these factors is used to generate a global output for the control of the actuators of the robots. Each robot maintains the formation by calculating its proper position in the group and executes a motor schema to move toward the goal position. Some important task allocation methodologies for MRS are presented in the following sections.

### 2.4.1 Strategies scenario

*A. Functionally-Accurate Cooperative (FA/C) distributed problem solving*

In the FA/C distributed problem solving approach presented by Lesser [36], each robot in the group has just partial data for solving the imperfect and temporal sub-problems. The FA/C paradigm provides an architecture for dealing with the situations where i) agents are solving mutually dependent, large-grained sub problems; ii) agents can generate partial and tentative high-level solutions in spite of incomplete and uncertain information; and iii) agents can partly resolve inconsistencies and uncertainties based on constraints derived from partial solutions to interdependent subproblems received from other agents.

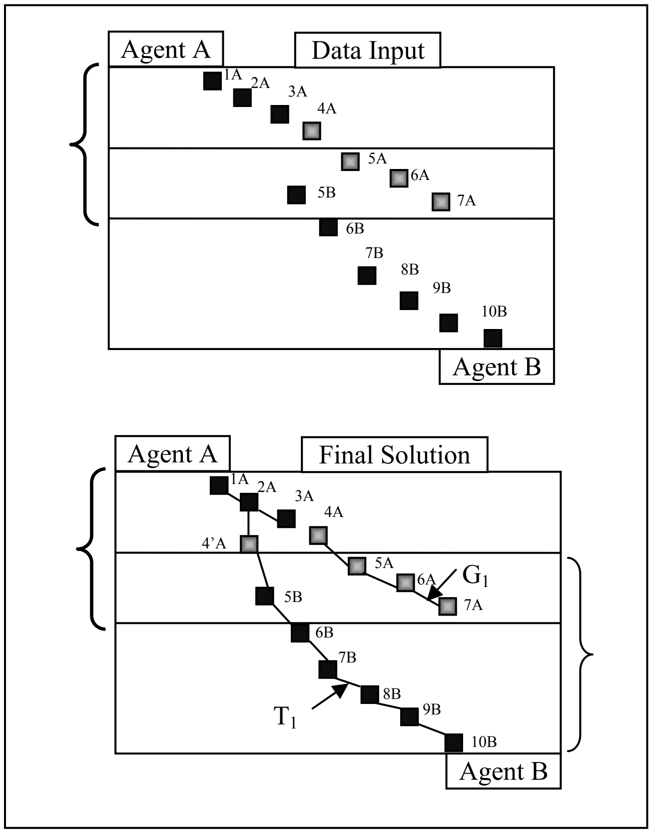



Figure 2.1 An example of a two-agent distributed aircraft monitoring scenario

The focal point of the approach is on the solution and control uncertainties that occur when a search is partitioned between agents and examines this concern from the conceptual viewpoint of a goal-based search and from the more practical viewpoint of a distributed interpretation of task. The occurrence of considerable amount of solution and control uncertainties in agents' local searches gives rise to uncoordinated behavior among the agents. The author describes a chain of increasingly sophisticated mechanisms for decreasing these uncertainties with the consequent increase in the coherence of agent activities. They comprise of integrating data and goal-directed control, using static metalevel information specified by an organizational structure, and using dynamic metalevel information as developed in the limited global planning structure. Each of these mechanisms provides information that reduces solution and control uncertainty. The structure of the two-agent distributed aircraft monitoring is presented in Figure.2.1.

*B. Alliance*

The ALLIANCE approach [37, 38, 39, 40, 41] is focused control architecture, ALLIANCE, that was developed essentially to facilitate fault tolerant, reliable, and adaptive cooperation among small- to medium-sized teams of mobile robots, performing in dynamic environments. ALLIANCE is a completely distributed, behavior-based architecture that incorporates mathematically-modeled motivations within each robot to achieve adaptive action selection. This architecture assumes a heterogeneous team of robots. A powerful force in the improvement of robotic systems is their prospective for reducing the need for human occurrence in dangerous applications.

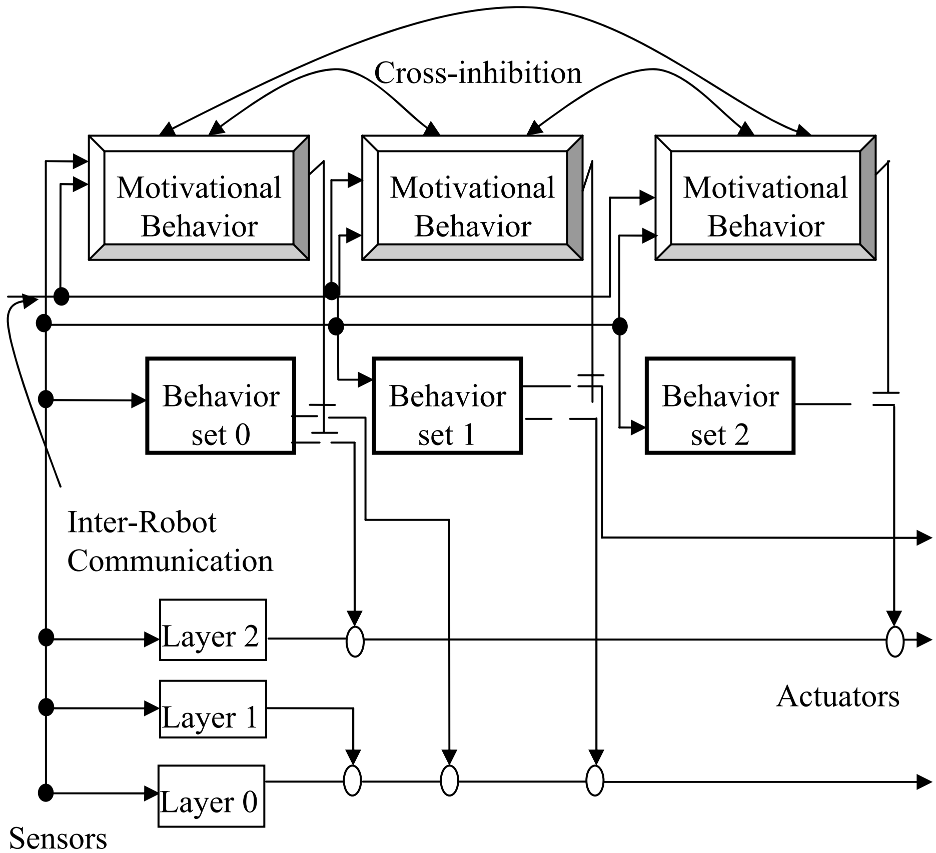



Figure 2.2 The alliance architecture

Applications such as the cleaning of toxic waste, nuclear power plant decommissioning, planetary exploration, fire fighting, search and rescue missions, security, surveillance, and reconnaissance tasks have elements of danger in which risks to operator are possible, or even likely. In all of these applications, it is desirable to decrease the risk to humans through the use of autonomous robot technology. Every robot just wants to run an ALLIANCE process as a requirement in order to assist. The ALLIANCE architecture is shown in Figure 2.2. A comprehensive approach, which incorporates learning, is called L. Alliance [42, 43]. Extensions to this approach are essential, however, when a robot must choose among a number of challenging actions-actions which cannot be pursued in parallel. Unlike characteristic behavior-based approaches, ALLIANCE delineates more than a few behavior sets that are either active as a grouping or are hibernating. It is attention-grabbing to note down that with certain restrictions on parameter settings, the

ALLIANCE architecture is assured to allow the robot team to complete its assignment for a wide variety of applications.

*C.Task acquisition using multiple objective behavior coordination*

Pirjanian [44, 45] presented a task allocation approach for deliberative behavior-based architecture for MRS. It is demonstrated that multiple objective assessment theory provides an appropriate formalism to cover thoughts from behavior based system synthesis and control, where each behavior is cast as an objective function estimator. Action selection comprises of generating and then selecting a set of pleasing solutions amongst a set of solutions that are Pareto-optimal. The basic thoughts of the planned methods are demonstrated through a set of simulated as well as real world experiments. Multiple objective decision making provides approaches to making decisions in difficult situations where more than one decision objective should be considered. By considering all system objectives concurrently these methods facilitate a smooth blending of several behaviors. However the investigational studies cast light on a most important problem namely deadlocks. It is extremely significant to deal with the deadlock problem in a structured manner.

*D. Team formation-based task allocation*

Stone and Veloso [46] introduced periodic team synchronization domains, as time-critical environments in which agents act autonomously with limited communication, but they can periodically synchronize in a full communication setting. They present a team agent structure that allows for an agent to capture and reason about team agreements and achieve collaboration between agents through the introduction of formations. A formation decomposes the task space defining a set of roles. Homogeneous agents can flexibly switch roles within formations, and agents can change formations dynamically, according to predefined triggers to be evaluated at runtime. This flexibility increases the performance of the overall team.

*E. Murdoch: publish/subscribe system*

Murdoch, a dynamic task allocation mechanism using a communication method called publish/subscribe is presented by Gerkey, and Mataric [47, 48] presented by for performing distributed control and multi-robot coordination. Multi-robot coordination is a complex control problem, particularly in tightly-coupled tasks that involve a mutual confidence of the robots on each others' performance. Thus tasks are divided at the behavior abstraction level instead of robot abstraction level. For instance, a task requiring sonar, laser, and vision publishes using the tuple of sonar laser camera as shown in Figure 2.3 to push the box along the desired trajectory. The problem is even more difficult through the use of heterogeneous robots, with different capabilities. The scheme of the mechanism can be described as follows. The robots are not equipped with gripping devices, but instead move objects by pushing against them. The pusher robots have no global positioning information and cannot see over the object; thus a watcher robot has the responsibility for leading the team (and object) to the goal, which only it can perceive. The system is entirely distributed, with each robot under local control.

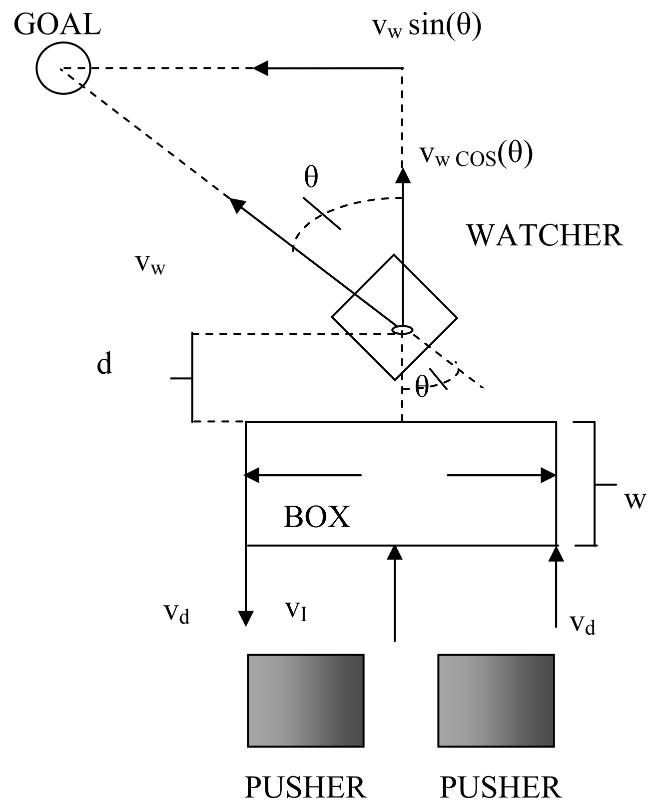



Figure 2.3 The model for deriving the pushing velocities for moving the box along the desired trajectory

A best-fit selection algorithm is used to choose the best among robots that are registered for a particular subject. The human user or another component of the system must perform task decomposition. Each task is accompanied with a metric as a measure of fitness. This metric is application-dependant and can be related to the robot's state or other computation. Afterwards, each registered robot measures its own fitness based on the metric and communicates the score to the others. The winner gains a time limit within which to accomplish the given task. This method finds its applicability where normal communication techniques are not possible because of complex heterogeneity in the pool.

*F. Robot exploration with combinatorial auctions*

Berhauld, Huang and Keskinocaki [49] proposed an appropriate for coordinating a team of mobile robots to visit a number of given targets in partially unknown terrain. Robotics researchers have studied single item auctions (where robots bid on single targets) to perform this exploration task but these do not take synergies between the targets into account. Therefore design of combinatorial auctions (where robots bid on bundles of targets), propose different combinatorial bidding strategies and compare their performance with each other, as well as to single item auctions and an optimal centralized mechanism. The results of Team Bots, a multi-robot simulator, indicate that combinatorial auctions generally lead to significantly superior team performance than single-item auctions, and generate very good results compared to an optimal centralized mechanism.

For the exploration tasks, robots are a natural choice for the bidders, and targets are a natural choice for the items. The auctioneer is a virtual agent who has sole responsibility for holding auctions and determining their winners but has no other knowledge and cannot control the robots. Initially, no robot owns any targets. Whenever a robot visits a target or gains more information about the terrain, it shares this information with the other robots and the auctioneer starts a new auction that contains all targets that have not yet been visited. The auctioneer could hold auctions less frequently or with fewer targets, but this would decrease the responsiveness of the robots to new information about the terrain. Each robot, including the current owner of a target, then generates bids in light of the new information and use sealed-bid single-round combinatorial auctions. Alternatively, multi-round combinatorial auctions that save bidders from specifying their bids for a large number of bundles in advance, and can be adapted to dynamic environments where bidders and items arrive and depart at different times. However, the auctioneer would then have needed to determine winners in every round and communicate some information about the current bids to the bidders, which would have increased the amount of computation and communication, respectively. The auctioneer closes the single-round auction after a predetermined amount of time, determines the winning bids, and notifies the winning robots. The winning bids are those that maximize the revenue of the auctioneer with the restriction that each robot wins at most one bundle per auction.

*G. Auction algorithm*

Bertsekas [50] presents an auction algorithm for task allocation in multi-robot applications. This is especially suitable for parallel computation. The auction algorithm is an intuitive method for solving the classical assignment problem. It outperforms substantially its main competitors for important types of problems. The assignment problem is important in many practical contexts.

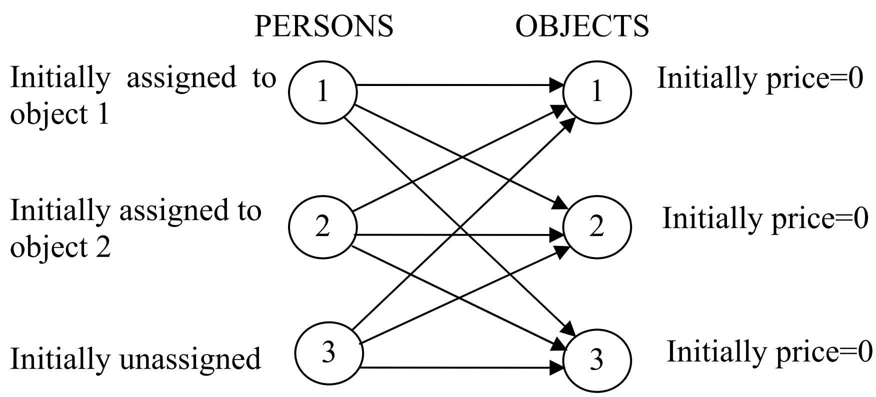



Figure 2.4 The structure of auction algorithm

The most obvious ones are resource allocation problems, such as assigning personnel to jobs, machines to tasks, and the like. There are also situations where the assignment problem appears as a subproblem in various methods for solving more complex problems. The assignment problem is also of great theoretical importance because, despite its simplicity, it embodies a fundamental linear programming structure. The most important type of linear programming problems such as the linear network flow problem can be reduced to the assignment problem by means of a simple reformulation. Thus, any method for solving the assignment problem can be generalized to solve the linear network flow problem. In fact this approach is particularly helpful in understanding the extension of auction algorithms to network flow problems that are more general than assignment. This approach attempts to find the most excellent assignment between tasks and users, while maximizing the total benefit. The structure of the auction algorithm is presented in Figure 2.4.

*H. A free market architecture for distributed control of MRS*

Stenz and Dias [51, 52] implement task allocation as a free market system. The coordination of a big group of robots to resolve a particular task is a complicated problem. Centralized approaches can be computationally intractable, brittle, and insensitive to alter. Distributed approaches are not as prone to these problems, but they can be extremely sub-optimal. This is a novel approach for coordinating robots based on the free market system. Market economies are a proven way to systematize a large number of individuals into a creative group. The free market approach defines profits and price functions across the probable strategy for executing a particular task. The task is accomplished by separating it into sub-tasks and allowing the robots to offer and discuss to bring out these sub-tasks. Cooperation and competition emerge as the robots perform the task while trying to make the most of their personal profits. The consequence promises to be an extremely robust multi-robot team that can competently exploit resources and opportunistically deal with uncertainties in a dynamic environment. Considering a team of robots assembled to perform a particular mission, the objective of the group may be to execute the mission as well as to minimizing the costs. But it is not sufficient to describe just the income and price functions for the team. An example of the problem is presented in Figure 2.5.

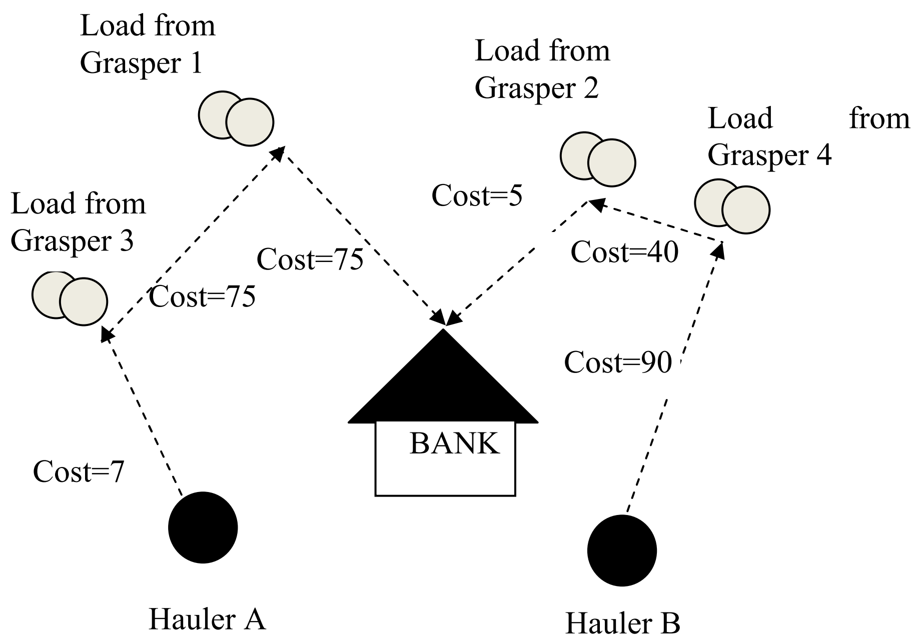



Figure 2.5 The winning TSP tour from robot A

*I. Broadcast of Local Eligibility (BLE) using Port Arbitration Behavior (PAB)*

Werger and Mataric [53, 54] present the Broadcast of Local Eligibility (BLE) mechanism the facilities comparison of locally determined eligibility for a given task with the best eligibility calculated by peer behaviors on other robots. When a robot's local eligibility is best for some behavior, it inhibits the peer behaviors on all other robots, and the task is awarded to it. In the case of robot or task failure, the resulting lack of inhibition will allow another robot to take over the task. Since BLE is based on broadcast messages to receiving ports that filter their input for the best eligibility, BLE-based systems are inherently scalable.

Initial

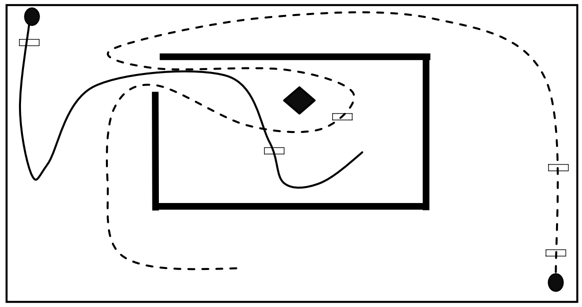


Figure 2.6 Initial assignments and final tours for 2 robots and 8 cities

The broadcasting technique uses a set of well-defined abstractions and techniques for behavior interaction and this is referred to as port attributed behavior (PAB) paradigm. When the PAB paradigm is extended across networks, the resulting systems are able to dynamically reconfigure themselves in order to optimally allocate resources in response to changing environmental conditions, in a manner that is scalable and robust to robot failures. The scheme of the process is shown in Figure 2.6.

### 2.4.2 Task allocation methodologies

*A. Distributed multi-robot task allocation (MRTA) for emergency handling*

Ostergaard and Mataric [55] describe a new prototype task, emergency handling, for multi-robot coordination. The experiments reported by the authors are to measure the effects of individualism and opportunism in a physically-implemented MRS. The authors use sound at multiple frequencies to simulate emergencies by producing several locally-sensable gradients in the environment. The results show that opportunism affords a significant performance improvement over individualism. The experiments also demonstrate the viability of sound for producing detectable local gradients in the environment. The scheme of the process is shown in Figure 2.7.

**Definition 1: The Emergency Handling**

Task consists of:

- An environment, *E*
- A set of robots, *R*
- A set of alarms, *A*
- A set of tools, *T*
- A capability function, c: $R \rightarrow T$
- A requirement function, $s : A \rightarrow T$

One robot can carry $|c(r_j)|$ tools, where $0 \leq |c(r_j)| < |T|, 0 \leq i < |R|$.

Each alarm can require $|s\ (a_i)|$, $0 < |s(a_i)\ | \ \leq min(|R|,|T), 0 < i <|A|$ tools to be fixed and require that all alarms can be handled with one or more of the available tools. Robots are heterogeneous if they are equipped with different tools or have different capabilities.Otherwise, the robots are homogeneous.

Figure 2.7 The elements of emergency handling

*B. Ants algorithms*

The basic idea of Ants algorithm [56] is based on adaptability of groups of ants to their environment changes. The method is based on some biological facts about ants, where they leave some amount of pheromone on their trail, and they prefer to follow the paths with most pheromone on it. This approach can be considered as task allocation, since each path/trail can be thought of as a task which must be selected with a probability function. This methodology is based on a few assumptions, including the fact that ants walk in a direct path, moving in a two-dimensional dimension. Another assumption is that when a group of ants encounters an obstacle, they divide into two equal sub-groups. An important feature of this approach is the indirect communication between ants, resulting in emergent behavior. The flow chart of the single agent controller is shown in Figure 2.8.

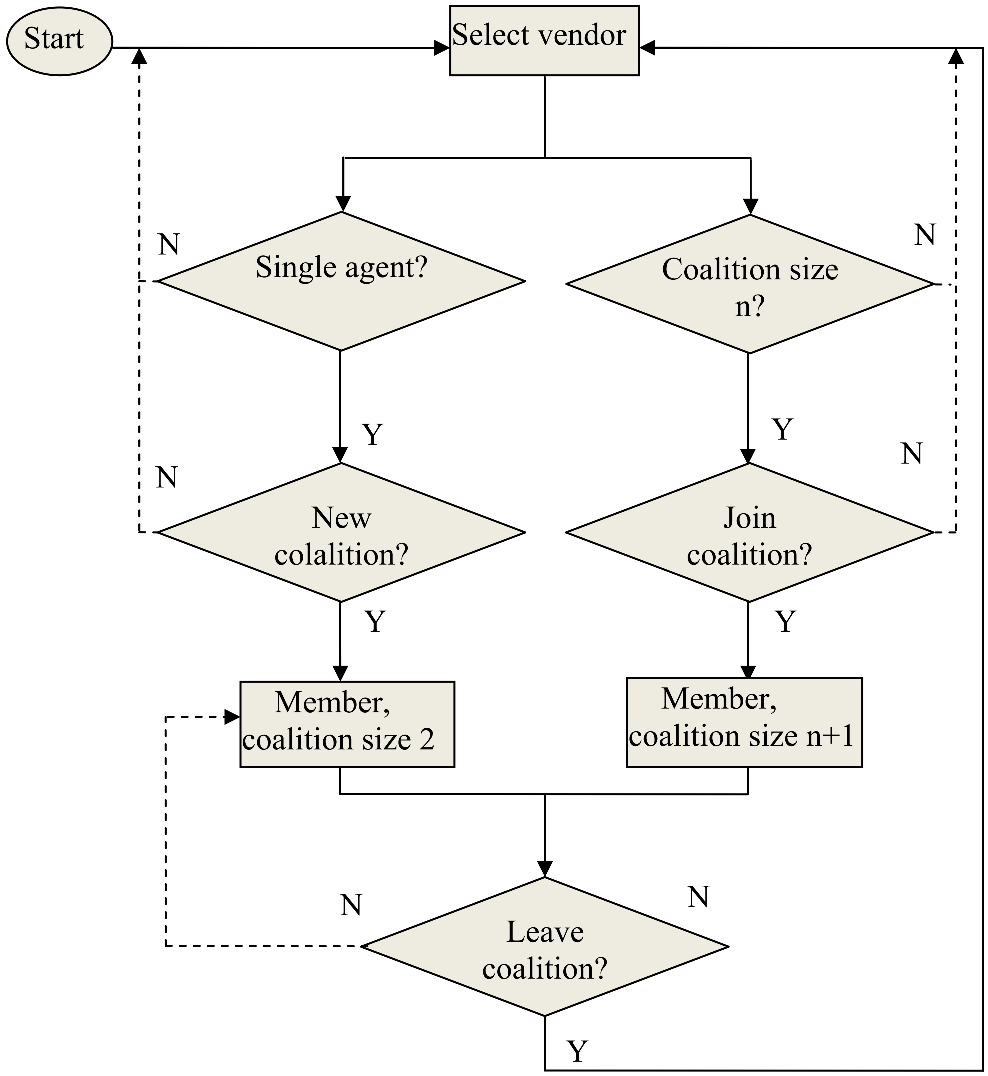



Figure 2.8 Flow chart for single agent controller

*C. Task allocation in uncertain environment*

Multiple cooperating robots hold the promise of improved performance and increased fault tolerance for large-scale problems such as planetary survey and habitat construction. Multi-robot coordination, however, is a complex problem. This problem in the framework of multi-robot dynamic task allocation under uncertainty has been described as an empirical study that sought general guidelines for task allocation strategies in MRS. Mataric et.al [57] identified distinct task allocation strategies, and demonstrates them in two versions of the multi-robot emergency handling task. An experimental setup has been presented to compare results obtained from a simulated grid world to those obtained from physical mobile robot experiments. Data resulting from eight hours of experiments with multiple mobile robots are compared to the trend identified in simulation. The data from the simulations show that there is no single strategy that produces best performance in all cases, and that the best task allocation strategy changes as a function of the noise in the system. The result is significant, and shows the need for further investigation of task allocation strategies and their application to planetary exploration.

*D. Cooperative task planning of MRS with temporal constraints*

Lian and Murray [58] discuss a design methodology of cooperative trajectory generation for MRS. The trajectory of achieving cooperative tasks, i.e., with temporal constraints, is constructed by a nonlinear trajectory generation (NTG) algorithm. The Advanced Highway Systems (AHS) and Mixed Initiative Control of Automa (MICA) hierarchies with their key elements and functions are shown in Figure 2.9.

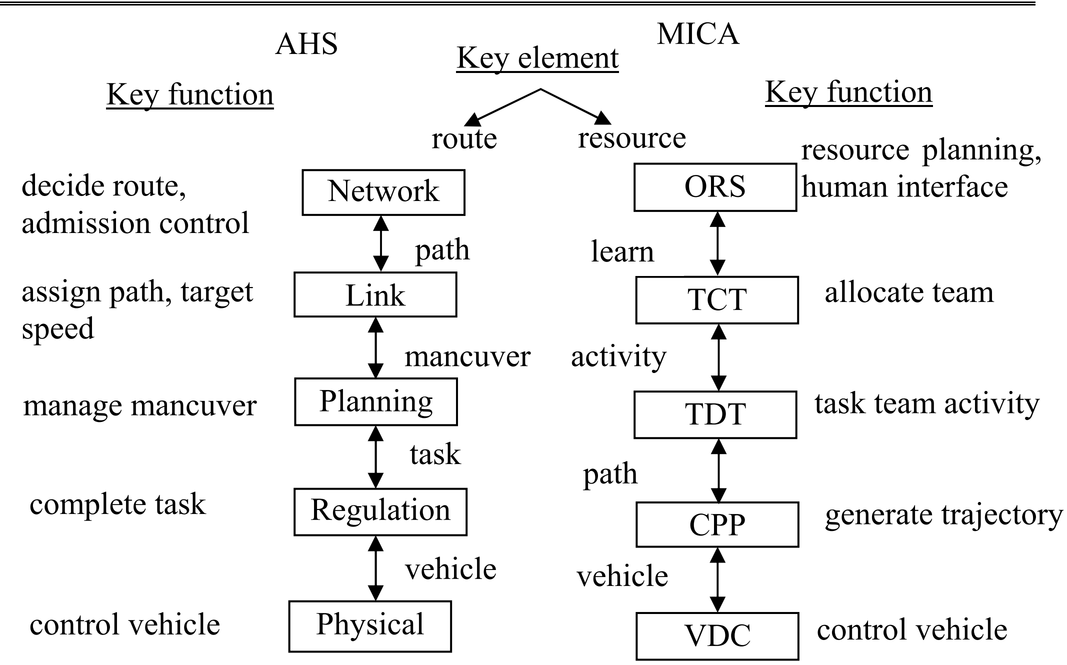



Figure 2.9 The AHS and MICA hierarchies with their key elements and functions

Conceptually, the MICA hierarchy includes Operations and Resources Supervisory (ORS) for resource planning and human interaction, Team Composition and Tasking

(TCT) for specifying group-level tasks, Team Dynamics and Tactics (TDT) for tasking team activities, Cooperative Path Planning (CPP) for generating feasible vehicle missions, and Vehicle Dynamics and Control (VDC).

There are three scenarios of robot tasking from home base to target position.

- A single robot is tasking from the home base position to the target position. The target position and the designated action at the position are simply instructed by an upper-level command unit.
- In the second case, three robots might be instructed by the same activity command, and need to move together in a designated formation. Hence, the controller at each individual robot should generate a set of feasible, real-time trajectories which guarantee the group of robot to move in the designated formation.
- The third case considers a more general scenario where multiple robots from different home bases are commanded to either one common target or multiple targets. At some location, these robots are commanded to move together and have a certain level of formation interaction. Conceptually, this scenario can be viewed as a combination of the first two cases.

For a given system dynamics and a set of state and input constraints, and to minimize a pre-specified cost function, the NTG algorithm first makes use of the differential flatness property to find a new set of outputs in a lower dimensional space and then parameterizes the outputs by the B-spline basis representation.

*E. Integer programming for combinatorial auction winner determination*

Andersson, Tenhunen and Ygge [59] recommend that on combinatorial auctions are important as they enable bidders to place bids on combinations of items; compared to other auction mechanisms, they often increase the efficiency of the auction, while keeping risks for bidders low. However, the determination of an optimal winner combination in combinatorial auctions is a complex computational problem. The authors compare recent algorithms for winner determination to traditional algorithms, and present and benchmark a mixed integer programming approach to the problem, which enables very general auctions to be treated efficiently by standard integer programming algorithms. The impact of the probability distributions chosen for benchmarking is discussed at length in their work.

*F. Physical interference impact in MRTA auction methods*

Guerrero and Oliver [60] opine that task allocation is one of the main problems in MRS. Among other factors, to get a good task allocation, taken into account the physical interference effects between robots, that is, when two or more robots want to access to the same point at the same time. They analyze interference impact using auction methods, one of the most popular task allocation systems. This approach shows how the performance of the auction utility function can be improved if interference impact is included in it and also provide a framework to simplify the method of finding a good utility function, which happens to be one of the major issues in all auction systems.

Classical auction methods have been modified to select which robots, and very specifically, how many of them are needed to execute a task. In an initial stage, each robot is looking for a task, and a robot finds a new task, it will try to lead it. There is only one leader for each task. If a robot is promoted to leader, it will create, if necessary, a work group; that is, a set of robots that will cooperate to execute this specific task. In that case, the leader must decide which the optimum group size is and what robots will be part of the group. To take this decision, the leader uses an auction like mechanism. During this process robots bid using their work capacity. The work capacity is the amount of work that a robot can execute per time unit, thus, this value is the utility function of our auction method. The leader selects the robots with the highest work capacity, until it detects that the group is able to reach its deadline, that is, until this condition is verified:

Also, in general, if the utility functions are not linear, the learning process can be very hard. To simplify the process, some parameters can be analyzed prior to the process, using an ideal environment, and then it can be modified during the execution of the task in the following 3 steps:

- Individual utility: during the first stage, evaluate the characteristics of each single robot without taking into account the others. Here it will include some characteristics like velocity, acceleration, etc.
- Group utility: in this step, the robot will take into account the other ones to create a coalition or working group. Here some parameters, like interference effect, will be included. That is, the robots will calculate the utility function of the group.
- Inter-Group utility: finally, the robots have to take into account that the decision of one group can affect to other groups. This inter-group dependency must be included in the utility function during the final step.

*G. Fuzzy multiple criteria assignment problems for fusion*

Gungor and Gunes [61] propose an assignment problems including multiple purposes and whose purposes featuring in a fuzzy way. In their work, 0-1 linear goal programming models of fuzzy multiple criteria assignment problems representing different-structured purposes are made up. Furthermore, Hungarian algorithm, is used for the solution of classic assignment problems obtained by changing $C_{ij}$ coefficients suitably according to fuzzy purposes in some fuzzy multiple criteria assignment problems. The objective of this approach includes a) to minimize total cost, b) to reduce the finishing time, c) to lower numbers of error, d) appointment of staffs numbered priority to the others, e) appointment of staffs numbered to machines numbered, f) appointment of staffs numbered to tasks numbered. The results of classical assignment problem formed by taking coefficients in the matrix into account with Hungarian algorithm have the same results as obtained by using linear goal programming model.

*H. Algorithm of task allocation based on realizing at the lowest cost in MRS*

L.Zu et.al [62] observes that the popular and several restricted forms of task allocation issue are NP problems. It searches a feasible matching scheme to realize corresponding object models. Then their approach adopted Hungarian algorithm to realize task allocation of the robots based on two-dimensional assignment problem aiming at multi mobile robot system. It resolves the problem for the robot how to get the tasks and realize them at minimal cost and designed an emulational test bed based on the multi-robot material flow system of the storages and docks which made distributed Programming using LAN. They also made some emulational experiments on Hungarian algorithm and compared it with the other algorithms.

*I. Combinatorial bids based MRTA method*

L. Lin and Z. Zheng [63] is conclude that coordinating several robots to cooperatively accomplish relatively complex tasks is not an easy issue. The author presents a combinatorial bids based MRTA method. An important basis to this technique is the capability category and capability vector formal description method. As the typical auction (or combinatorial auctions) based mechanisms have some inherent disadvantages, they propose a novel method: combinatorial bids based mechanism. This new method provides an explicit cooperation mechanism to the bidding robots so that they can form a subset to bid for complex tasks. Validation of this approach is based upon the Player/Stage system. They carefully designed a desk and chair moving scenario to test the algorithm and compare it with the typical auction based method. Robots and tasks are both highly heterogeneous embodied in their variant capability vector. Carefully designed simulations indicate that the combinatorial bids based method is more efficient than the typical auction based one.

*J. Optimal robot selection and work station assignment for a CIM system*

Jack and Bernard [64] use a mathematical program and solution algorithm develop an optimal robot selection and work station assignment for a computer integrated manufacturing system. In specific, the model considers selection of a proper mix of multiple-type robots such that operational requirements from a given number of work stations are satisfied at minimal system cost. Each robot is characterized by its fixed charge, and subject to limits on machine time and work envelope. Each work station has known demands on both robot machine time and work space. The model is formulated as a pure 0-1 mathematical program and is shown harder than two-dimensional bin packing, a well-known NP-hard problem. A three-phase optimization algorithm is implemented and tested by solving 450 randomly generated problems. Computational results indicate the solution algorithm is effective in solving problems of a practical size.

*K. Simulated annealing for multi-robot hierarchical task allocation with minmax objective*

Mosteo and L.Montano [65] study algorithms for minimizing the worst-case cost of any agent in a multi-robot team in time critical missions. They propose a generalized model for flexible mission planning, using hierarchical task networks as the planning framework, and the multiple traveling salesman problems as the cost model for task allocation. Two approximated solutions are provided and compared for this NP-hard problem, one based in current research in market-based techniques, and another one based in the optimization technique known as simulated annealing. The authors provide simulation results which back the model described and the proposed algorithms.

*L. Task assignment for a small batch flexible assembly cell incorporating multiple robots*

Boneschanscher [66] presents a task assigner for a Flexible Assembly Cell (FAC) incorporating multiple robots and a transport system. The FAC can assemble a wide range of products in small batches. Parts are fed on pallets and assembled on fixtures, which both can route through the cell. The FAC has a limited buffer capacity. The task assigner determines a schedule for each batch, with minimum assembly time as the main objective. Task assignment is done for a limited time horizon, using a goal directed search. The time horizon is determined by the limited buffer capacity of the FAC. While assigning tasks to resources in the cell, the task assigner determines an appropriate assembly sequence and allocates tools such as grippers to workstations in the cell.

Sahu and Tapadar [67] attempt to solve the generalized "Assignment problem" through genetic algorithm and simulated annealing. The generalized assignment problem is basically the "N men- N jobs" problem where a single job can be assigned to only one person in such a way that the overall cost of assignment is minimized. While solving this problem through GA, a unique encoding scheme is used together with Partially Matched Crossover (PMX). An experimental investigation into solving the Assignment model using GA and Simulated Annealing (SA) is presented.

Although many significant results have been obtained by the researchers in the area of MRS, a great deal of work remains to be done in order for the behavior of the team of robots and utilized in dynamic environment. The idea of task allocation remains a necessary component of this challenge. A survey of this field was included in this research work. Productive, efficient, and dynamic approaches to assignment of tasks to different robots will result in further utilization of multi-robot systems.

Task allocation and decomposition methodologies will serve as guidelines to allow MRS to gain efficiency. It is significant to spend time to realize different methods and apply in the different applications. Some progress has been made in the last few years to extend and apply MRTA in the dynamic environment. With further research, they should become more broadly applicable and more competitive with the single robot systems.

## 2.5 Summary

In this chapter, the various lines of work relevant to this thesis are introduced. A brief overview of research on strategies, selection of robots and task allocation approaches are provided in this chapter. These fields are too large to be covered adequately within the space of a few pages, but the review is broadly categorized on strategies, selection and task allocation of robots. The overall goal is to construct reliable strategies, select the suitable robots and optimized the task allocation to robots that are reasonably well-specified. All of the above research works have, in one way or the other, attempted to solve this problem.

# Chapter-III

## PROBLEM STATEMENT AND STRATEGIES

# CHAPTER 3

## Problem Statement and Strategies

### 3.1 Introduction

Robotic installations, for obvious reasons, are costly propositions at the first glance. There has been a very strong focus on efficient robot operation in industries in order to make the system economically competitive and responsive. These issues, especially in MRS are too sensitive to make it feasible for cutting edge industries. However, the system and the application scenario should be thoroughly understood before creating the MRS.

In order to make an MRS work as per the need and to make it efficient and competitive with available resources, there ought to be certain design and operational strategies. These strategies can be developed/selected and implemented to make the system work in the best possible manner. Strategies, as such, do change with the situations encountered and can be different even for the same situations. The following sections present the focused area of the research work and the various possible strategies that may be adopted in the envisaged MRS.

A significant challenge in many dynamic multirobot application domains is the lack of complete and reliable information. Coordination strategies need to be a lot more flexible if all information is not known a-priori. In certain domains, often much is unknown about the prevailing conditions of the environment. Hence, robots need to rely on their sensors to discover these conditions. Thus, the information will only be good as the sensing capability and thus uncertainty is introduced.

The dynamic nature of the environment further exacerbates the challenge since discovered information cannot be relied on as perfect or sustained. A successful coordination mechanism needs to take all of this into account and deal with the challenges of imperfect information in an efficient manner. Some of the real life situations may be as follows.

- Industries have variety of tasks to perform either at the same time or at different time thereby requiring flexible automation agents to assist in functioning.
- Some tasks require multiple skills and capabilities whereas some tasks may require multiple agents' simultaneous or cooperative effort for its accomplishment
- Sometimes multiple types of agents are required to enhance throughputs and efficiency of the system.
- Selection of candidate robots should be based on the task needs as well as the economic consideration of the system in order to make the system commercially viable and operationally efficient.

The aforementioned industrial situations attract a good number of objectives to be handled at the same time. Some of the objectives can be precisely stated and quantified whereas some are sparsely stated. In order to handle the problems to the benefit of industries and users it is important that the problem is looked at with a broader and strategic perspective.

## 3.2 Problem Statement

After going through the related literature in the focused area and in the context of the broad objective as mentioned in section1.10, the problem for the present work is described in the following lines.

i) To identify certain areas of industrial operations where multiple robots are/can be employed to enhance the productivity and system efficiency.
ii) To adopt /develop strategies for deployment of multiple robots for industrial applications.
iii) To develop methodology(ies) in a systematic and scientific manner for selection of appropriate type(s) of robots for the intended application and to recommended the suitable one for a specific situation.
iv) To explore and develop various task allocation procedures in MRS with different operating conditions and resource types with a view to minimize total cycle time and with better utilization of the resources.
v) To recommended appropriate methodologies for selection of robots and assignment of tasks to the candidate robots under various working conditions and for different problem sizes.

Since a number of strategies, selection methods for robots and assignment rules are applied to solve the problem so envisaged, a comparative study is necessiated.

## 3.3 Strategies for accomplishing tasks by robots

Strategies for manipulating objects in our everyday life are adopted. While screwing the cover onto a jar, usually one holds the jar with one hand and the cover with the other. It may wiggle the cover if it is detected that the jar's axis of symmetry is not aligned with that of its cover, or to think that the threads are not matching right. A similar process occurs when one tries to insert a key into a lock, or whenever one tries to assemble two objects, one of which fits into the other.

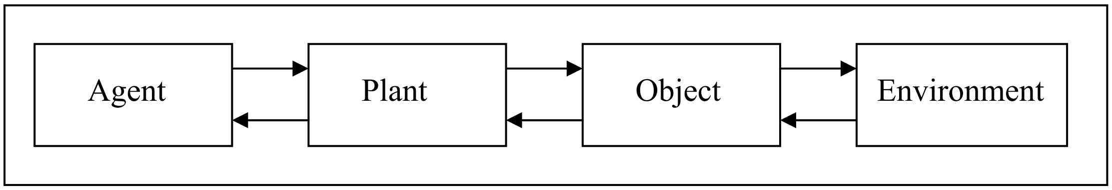



Figure 3.1 Abstract block diagram of MRS environment

Some strategies involve rigid objects while others may operate on flexible objects like shoe-laces or articles of clothing. Some, like the assemblies mentioned above, seem to involve a constant and complicated monitoring of forces and positions. Others work with less complex sensing. In fact, at one extreme, there are tasks where strategies, that seem to require no sensing at all, are used.

Figure 3.1 shows an abstract depiction of a typical task. The agent can be a human brain or a computer process. This agent interacts with and controls a plant - arms and legs in the former case, a robot manipulator in the latter. The plant, in turn, is assumed to be in contact with some objects which are being manipulated.

### 3.3.1 The design and analysis of strategies

There are two aspects of strategies that are quite important. The first involves the design of strategies to accomplish tasks in a given domain. The second involves the analysis of a strategy to understand its scope and applicability. The environment geometry may change between tasks. In some sense, it is this variation that requires using a strategy, by which to mean an abstract, generalized, parametrized description of what to do in order to accomplish a task. There are many important components of how strategies can handle variations, a few of which are mention below.

1.Manipulation strategies rely on planning to predict the future outcome(s) of an action. In task domains where models of the world and models of the interactions illustrated by the arrows in Figure 3.1 are accurate, planning could play an important role.

2. Strategies also involve sensing variables that relate to accomplishing a given task. If a strategy does not have access to accurate models which allow one to predict what might happen when an action is executed, then it must rely on sensing to find out. If one can sense such task relevant variables often enough, and exercise actions to guide the evolution of the task in the right way, then sensing can compensate for the lack of planning capability.

3. Strategies can also use task mechanics to accomplish a task successfully. Tasks that can be executed in a purely sensorless fashion, and special-purpose mechanisms that are built to execute a single task passively, illustrate that task mechanics can be exploited quite effectively in some cases.

4. Some strategies seem to rely on randomness to accomplish their goals in an expected (or average) sense. In some tasks, such strategies execute (on the average) faster than other strategies that prepare for the worst case and seek to produce guaranteed solutions.

### 3.4 Task allocation in MRS

To accomplish the desired tasks, it is required to judiciously plan and sense, exploit task mechanics where possible, and rely on randomness when guaranteed approaches fail. It is not a-priori clear, however, how to design strategies that involve trade-offs between all of these components. The analysis of strategies is a much more daunting task, especially when one considers the variations that must be taken into account. There are many interesting questions that can be asked regarding the scope and performance of a strategy. For example, one might be interested in how well a strategy handles uncertainty and whether it performs equally well in the face of small or large errors in control and sensing. The performance of the system under changing environment should also be a point of consideration. Handling uncertainty that cannot be predicted or taken care of during the design phase is also another important issue during the task accomplishment.

### 3.5 Types of strategies

There can be number of strategies to handle real life problems. Some of the strategies that are considered for the present MRS are:

i) Task allocation strategy, and
ii) Robot operation strategy

#### 3.5.1 Task allocation strategies

There can be a number of strategies for task allocation amongst a number of robots constituting the MRS with a given goal. The task allocation can be made depending upon the types of robots available, type of tasks to be carried out, nature of operation to be performed, economic considerations, if any, and time of completion as per target. One of the following strategies can be advantageously adopted for situations in place.

i) motivation-based,
ii) mutual inhibition,
iii) team consensus,
iv) no allocation, and
v) auction-based.

The first strategy, motivation-based task allocation, uses an internal motivation mechanism to cause behavior changes. Parker’s ALLIANCE [68] and stagnation recovery by Kube and Zhang [69] are two best-known examples. Motivation-based cooperation distributes the task allocation process equally among members of the team, and emergent team behavior results from simple control mechanisms within each agent. For example, an agent may be triggered to change its own task assignment because excessive time has elapsed without task progress.

In task allocation through mutual inhibition, the second strategy, robots directly inhibit those around them from being chosen for a task, as in Werger and Matari´c’s Broadcast of Local Eligibility (introduced in [70] and expanded in [71]) and emergency handling by Ostergaard, Matari´c, and Sukhatme [72]. These mutual inhibition techniques require regular broadcasts by each robot communication overhead for m tasks and n robots and a shared global representation of the available tasks. Mutual inhibition is not considered viable for task allocation with low communication costs.

The third strategy, task allocation by team consensus, enables entire teams of robots to agree on a team strategy or formation. This has been used by Stone and Veloso to coordinate teams for RoboCup [73]. Jones and Matari´c have explored multi-robot coordination where robots use only their internal state with no communication [74] [75]. Thus, the robots coordinate through the use of shared (but locally derived) models. The team consensus approach is not considered as a solution to the recruitment problem for two reasons: it relies on agents modeling the other agents in what may be a dynamic team, and there is no explicit call for help to begin the recruitment process.

Some approaches use a fourth type of strategy, no allocation, to coordinate robot teams, and it is assumed that all robots cooperate on the same task.

In the fifth strategy, auctions, robots explicitly negotiate for tasks through a bidding process. A common approach to auctions is the Contract Net Protocol (CNP, introduced in [76] and [77]) with a first-price auction. In CNP, an announcement about a new task is broadcast to a team of robots. Each robot then returns a bid that specifies how well-suited it is for the task. A winner is selected from the bids; in the case of a first-price auction, the bid with the best utility (or lowest cost) is chosen. Auction-based approaches allow agents in the team to maximize utility or minimize cost that results from the task assignment.

### 3.5.2 Task assignment in MRS through auction

An auction is a process of buying and selling goods or services by offering them up for bid, taking bids, and then selling the item to the winning bidder. In economic theory, an auction may refer to any mechanism or set of trading rules for exchange. There are several variations on the basic auction form, including time limits, minimum or maximum limits on bid prices, and special rules for determining the winning bidder(s) and sale price(s). Participants in an auction may or may not know the identities or actions of other participants. Depending on the auction, bidders may participate in person or remotely through a variety of means, including telephone and the internet. The seller usually pays a commission to the auctioneer or auction company based on a percentage of the final sale price. The different types of auctions are as follows:

- English auction is also known as an open ascending price auction. This type of auction is arguably the most common form of auction in use today. Participants bid openly against one another, with each subsequent bid higher than the previous bid. An auctioneer may announce prices, bidders may call out their bids themselves (or have a proxy call out a bid on their behalf), or bids may be submitted electronically with the highest current bid publicly displayed. In some cases a maximum bid might be left with the auctioneer, who may bid on behalf of the bidder according to the bidder's instructions. The auction ends when no participant is willing to bid further. Alternatively, if the seller has set a minimum sale price in advance (the 'reserve' price) and the final bid does not reach that price the item remains unsold. Sometimes the auctioneer sets a minimum amount by which the next bid must exceed the current highest bid. The most significant distinguishing factor of this auction type is that the current highest bid is always available to potential bidders. The English auction is commonly used for selling goods, most prominently antiques and artwork, but also secondhand goods and real estate.

- Dutch auction is also known as an open descending price auction. In the traditional Dutch auction the auctioneer begins with a high asking price which is lowered until some participant is willing to accept the auctioneer's price. The winning participant pays the last announced price. The Dutch auction is named for its best known example, the Dutch tulip auctions. In addition to cut flower sales in the Netherlands, Dutch auctions have also been used for perishable commodities such as fish and tobacco. In practice, however, the Dutch auction is not widely used.

- Sealed first-price auction is also known as a first-price sealed-bid auction (FPSB). In this type of auction all bidders simultaneously submit sealed bids so that no bidder knows the bid of any other participant. The highest bidder pays the price they submitted. This type of auction is distinct from the English auction, in that bidders can only submit one bid each. Furthermore, as bidders cannot see the bids of other participants they cannot adjust their own bids accordingly. This kind of bid produces the same outcome as Dutch auction. Sealed first-price auctions are commonly used in tendering, particularly for government contracts and auctions for mining leases.

- Vickrey auction, is also known as a sealed-bid second-price auction. This is identical to the sealed first-price auction except that the winning bidder pays the second highest bid rather than their own. This is very similar to the proxy bidding system used by eBay, where the winner pays the second highest bid plus a bidding increment (e.g., 10%). Although extremely important in auction theory, Vickrey auctions are rarely used in practice.

*The auction algorithm*

The auction algorithm is an intuitive method for solving the classical assignment problems. It outperforms substantially its main competitors for important types of problems, both in theory and practice, and is also naturally well suited for parallel computation. In the process, the user submits jobs to the auctioneer to start the process. An auctioneer is responsible for submitting and monitoring jobs on the user's behalf. The auctioneer creates an auction and sets additional parameters of the auction such as job length, the quantity of auction rounds, the reserve price and the policy to be used. The auctioneer informs the robots (Robot-1, Robot-2 and Robot-3) that an auction is about to start. Then, the auctioneer creates a call for proposals, sets its initial price, and broadcasts calls to all the robots (Robot-1, Robot-2 and Robot-3). Robots formulate bids for selling a service to the user to execute the job.

The robots evaluate the proposal; they decide not to bid because the price offered is below what they are willing to charge for the service.

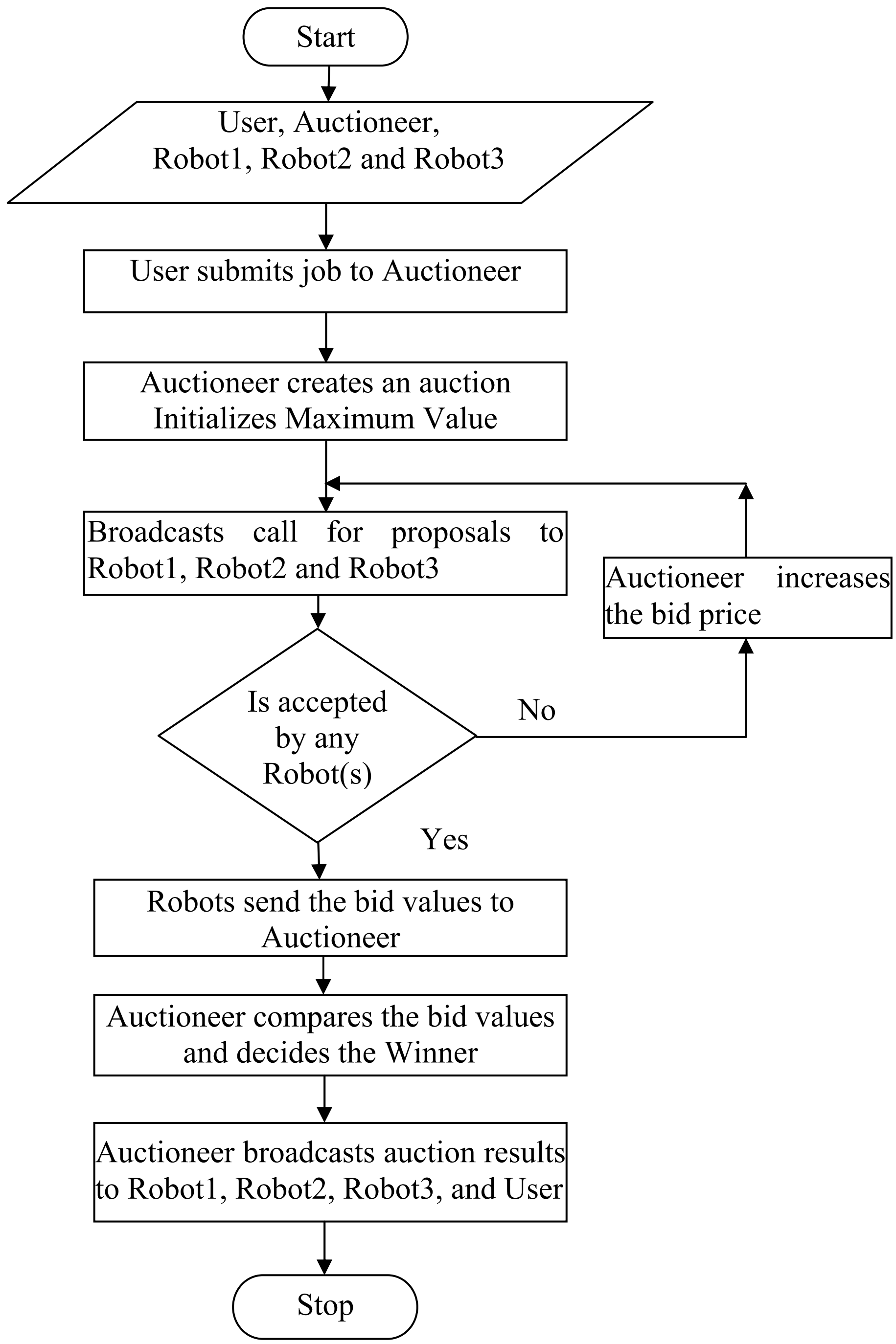



Figure 3.2 Flowchart of the auction for task

This makes the auctioneer to increase the price and send a new call for proposal with this increase in the price. Meanwhile, the auctioneer keeps updating the information about the auction. In the second round, Robots are decided to bid. The auctioneer clears the auction according to the policy specified beforehand. Once the auction clears, it informs the outcome to the user and the robots. The flowchart for the process is presented in Figure 3.2.

The auction algorithm includes the following steps:

In general cases, the auction proceeds in five steps

1) Task announcement.
2) Evaluations. Each candidate evaluates the cost and gain to execute the task then determine its bids.
3) Bid submission. Each candidate publishes its “score” representing task-specific fitness to the auctioneer.
4) Close of auction. The auctioneer processes the bids, determines the winner, and notifies the bidders.
5) Progress monitoring /contract renewal.

Such algorithm can deal with task allocations when the environmental information is only partly known and the failure of some candidates is tolerable. The method can be modified to fit multi-task allocation problem. However, the tasks may change in a dynamic environment so that even single-item-multi-round auction algorithms and combinatorial auction are not suitable for the situations in which the tasks change rapidly, for example, the cooperative hunting for a high-speed target. The hunters might arrive to a former assigned destination whereas the target had already wandered to some other places.

*Auction algorithm for task allocation*

1. An auctioneer that discovers a task commences an auction by announcing the task’s location and requesting bids from other robots.
2. Every robot that is within communication range of the auctioneer robot receives the information about the task being auctioned.

3. A robot that receives information about the task being auctioned can respond with a bid if only if it has at most one existing task in its task list. The value of the bid is given by the sum of straight line distances from the robot's current location to the auctioned task's location, via the location of the task, if any, on the robot's task list.
4. The auctioneer continues to receive bids till the time for that auction expires. The auctioneer robot then selects the top n bidders (*n* of the closest robots to the task) as the auction's winners. If the auctioneer receives *m* bids, where $1 \leq m < n$, it selects only *m* winners. If the auctioneer does not receive any bids it restarts the auction.
5. The auctioneer informs the winning robots that they were selected to perform the task, while the robots that lost the auction are informed that they were not selected to perform the task. The winning bidder with the lowest valued bid (robot furthest from the auctioneer) is informed by the auctioneer that it is going to be the last robot to visit the task for the current auction.
6. Each selected robot visits the task to partially complete it and deposits pheromone at the location corresponding to the task. The updated pheromone value is communicated to other robots that have been selected to perform the task, but have not yet performed the task.
7. A task is considered complete when the amount of pheromone associated with it reaches a threshold value of $\tau$.
8. If the last robot visiting the task observes that the pheromone value of the task is $< \tau$ after it has executed the task, it starts another round of auction for the same task. On the other hand, if the last robot observes the pheromone value of the task is $> = \tau$, it considers the task to be completed. The robot then communicates the task completion information for the task at that location to other robots within its communication range. This prevents robots from rediscovering the same task and initiating another auction for the task later on.

The algorithm described here can be utilized in task allocation in multi-robot applications, and is particularly suitable for parallel computation. This approach attempts to find the best assignment between tasks and robots, while maximizing the total benefit. It iterates between robots and in each iterations tries to assign a task to a robot who offers the most. In consecutive iterations, other robots may bid for other tasks and if more than one bid is available for the same task, it will increase the cost of task until finally just one task-robot pair match takes place, (iterative improvement). The iteration terminates when all robots are pleased with their match, otherwise an unhappy robot will bid higher for another task and this process will continue. Although auction algorithm may have some similarities to the free market approach, there is a little difference. One difference is that in the free market approach, agents can cooperate in order to gain a maximum profit for all of them, however in the auction algorithm every robot is considered rival. The auction algorithm uses an exclusive mathematical model for all the applications, while the free market approach does not. In addition, the free market technique is based on the collection of heterogeneous agents, while in the auction algorithm the robot set is homogeneous.

### 3.5.3 Robot operation and dynamic task assignment

The dynamic task allocation problem, i.e., the mapping from bids to tasks, can be performed in numerous ways. The focus is limited here to Markovian systems, where the task allocation mapping for a given robot is based on the mapping between that robot's current task assignments and every other robot's current bid on each task, to the given robot's new task assignment, as shown in Table 3.1. Given each robot's bid on each task, and its current task engagement, the new task assignment of each robot is required to be determined. Given the large space of possibilities, only the extreme cases of no commitment and full commitment, and no coordination and full coordination for each of the robots are considered. The combination of these extremes results in four task allocation strategies as shown in Table 3.2. Along the commitment axis, a fully committed strategy meant a robot would complete its assigned task before considering any new engagements, while a fully opportunistic strategy allowed a robot to drop an ongoing engagement at any time in favor of a new one. Along the coordination axis, the uncoordinated (individualistic) strategy meant each robot performed based on its local information, while a coordinated strategy simply implemented mutual exclusion, so only one robot could be assigned to a task, and no redundancies were allowed. It is noted that this notion of coordination is simple, and it is not intended to represent explicit cooperation and coordination strategies (i.e., the fixed time-cost was 0). During the process three new tasks appear every twelve time-steps at random positions on the grid. The tasks are structured so that one robot is sufficient for completion of an individual task assignment.

Table 3.1 An example of task allocation scenario

| Current engagement | Bids | A | B | C | D | New engagement |
|---|---|---|---|---|---|---|
| A | Robot-1 | 6 | 4 | 2 | 5 | ? |
| -- | Robot-2 | 4 | 1 | 0 | 3 | ? |
| C | Robot-3 | 7 | 2 | 3 | 2 | ? |

Table 3.2 The task allocation strategies

| Commitment 🡇 | Coordination 🡆 | |
|---|---|---|
| | Individual | Mutually exclusive |
| Commitment | Strategy-1 | Strategy-2 |
| Opportunity | Strategy-3 | Strategy-4 |

Thus, mutual exclusion is the simplest yet effective form of coordination. As an example, the fully committed mutually exclusive strategy is as follows:

1. If a robot is currently engaged in a task, and its bid on that task is greater than zero, remove the row and column of the bid from the table, and set the robot's new assignment to its current one.

2. Find the highest bid in the remaining table. Assign the corresponding robot to the corresponding task. Remove the row and column of the bid from the table.
3. Repeat from step 2 until there are no more bids. In case of individualistic (uncoordinated) strategies, the same algorithm is run on a separate table for each robot. In the opportunistic (uncommitted) case, step 1 above is skipped.

In the context of multi-robot coordination, dynamic task allocation can be viewed as the selection of appropriate actions [78] for each robot at each point in time so as to achieve the completion of the global task by the team as a whole. From a global perspective, in multi-robot coordination, action selection is based on the mapping from the combined robot state space to the combined robot action space. For homogeneous robots, it is the mapping;

$$S^{|R|} \rightarrow A^{|R|}$$

Where, S is the state space of a robot, |R| is the number of robots, and A is the set of actions available to a robot [79]. In practice, even with a small number of robots, this is an extremely high-dimensional mapping, a key motivation for decomposing and distributing control.

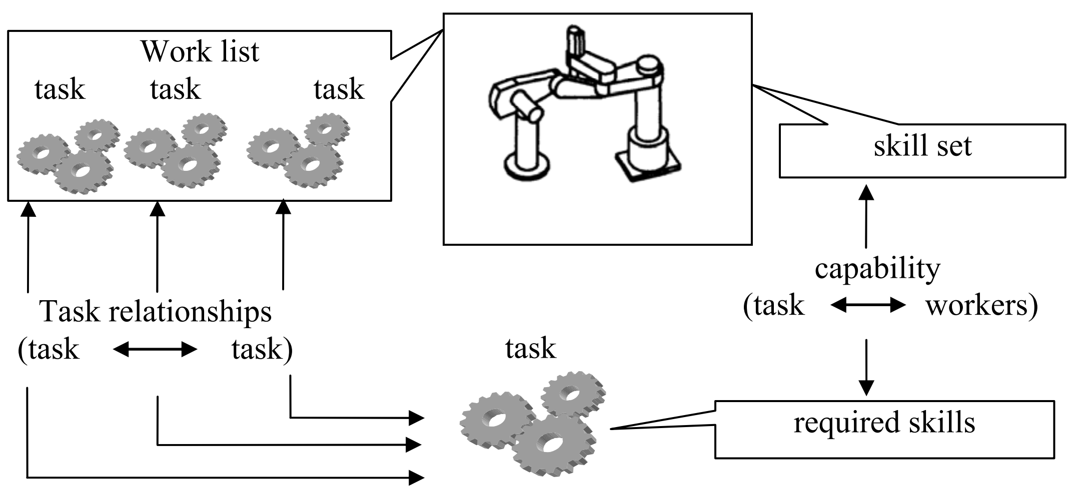



Figure 3.3 Evaluation criteria

Based on the approach introduced in [80], the task allocation problem is decomposed into the following three steps:

1. Each robot bids on a task based on its perceived fitness to perform the task;
2. An auctioning mechanism decides which robot gets the task;
3. The winning robot's controller performs one or more actions to execute the task.

The above decomposition is aimed at constructing a general formulation for the multi-robot coordination problem. In this formulation, a bidding function determines each robot's ability to perform a task based on that robot's state. Next, the task allocation mechanism determines which robot should perform a particular task based on the bids. Finally, the robot controllers determine appropriate actions for each robot, based on the robot's current task engagement. This partitioning, as illustrated in Figure 3.3, serves two purposes: it reduces the dimensionality of the coordination problem, and it reduces the amount of inter-robot communication required. We now have the mapping

$$B^{|R||T|} \rightarrow T^{|R|}$$

Instead of mapping, namely from all robots' bids B for all tasks T to a task assignment for each robot, this overall mapping is called the task allocation strategy for the system as a whole. The overall mapping is treated here as a global, centralized process (as depicted in Figure 3.4), but distributed auctioning mechanisms [81,82], blackboard algorithms [83], and cross-inhibition of behaviors [84] are some validated methods for distributing the task allocation function. In this methodology, the focus is on what the task allocation function should be, rather than on how it should be distributed. The above framework is a general way that dynamic task allocation for MRS can be formulated.

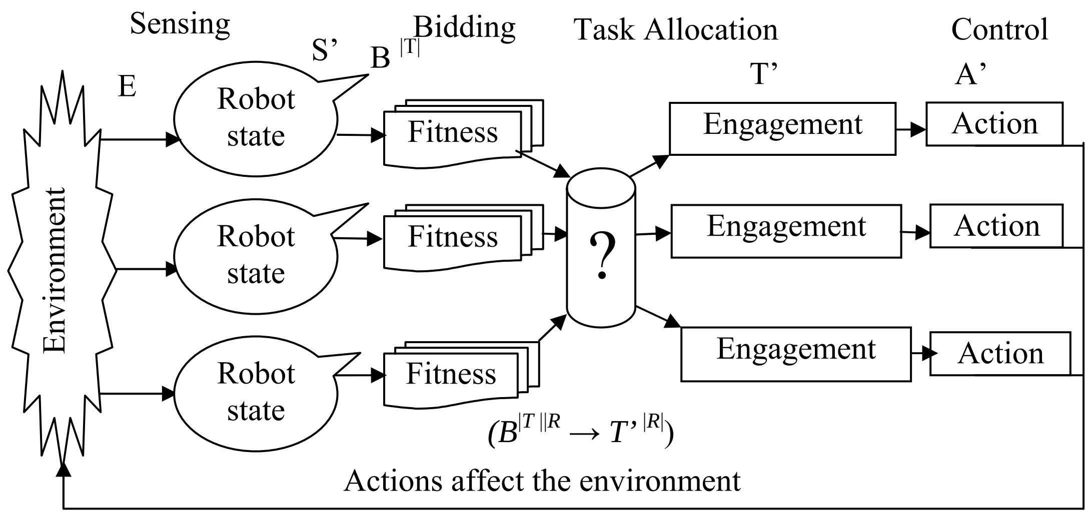



Figure 3.4 Reducing the dimensionality of multi-robot coordination

### 3.6 Summary

In this chapter mathematical models are developed for creating strategies for the robots. Considering the environment of tasks and type of robots in mind different strategies are identified for task as well as for robots. Finally the suitable strategies are sorted out for robots. An auction based algorithm is discussed in this chapter for allocation of robots to the tasks. A procedure for auction algorithm is outlined and the flowchart is presented.

# Chapter-IV

# SELECTION OF ROBOTS

# CHAPTER 4

## Selection of Robots

### 4.1 Introduction

Recent developments in information technology and engineering sciences have been the main reason for the increased utilization of robots in a variety of advanced manufacturing facilities. Robots with vastly different capabilities and specifications are available for a wide range of applications. The selection of robots to suit a particular application and production environment from among the large number of robots available in the market has become a difficult task. Various aspects such as product design, production system, and economics, need to be considered before a suitable robot can be selected. The selection problem is particularly relevant in view of the likely lack of experience of prospective users in employing a robot. Indeed, robots are still a new concept in industry as a whole, and so it is not unusual for an industry to be a first-time robot purchaser.

With the advancement of technology, production systems are changing from traditional human dependent systems to intelligent automated systems. Industrial Robots have been instrumental in making the production systems more efficient, productive, responsive and flexible. In large production systems, multiple robots of different types, capacities and capabilities are employed for accomplishing the desired tasks. The flexibility and scalability of the system is greatly enhanced by use of multiple types of robots. The concept of using multiple robot types comes from the availability of those robots in the market. However the use of multiple type robots in a single workcell should not be done in random manner. It is desired that all the, devices in a workcell are controlled and coordinated properly through a single point (host) so that the workcell behaves like a single entity. Hence it is important to have the compatibility of the robots with the host. This calls for robot selection for the intended workcell. Since a multirobotic workcell is a cost intensive proposition the planning of such workcell should be done correctly. The selection of robots and subsequently the allocation of these robots for accomplishing the goal become prime issues in making the system efficient both from operation and economy view points. There is good number of tools available for optimizing the general allocation problems. However, if the robots under consideration are in large number possessing higher capability and the number of tasks to be carried out is large, then the number of alternatives for allocation becomes exorbitantly large, thereby making the allocation problem an NP-hard. Therefore, the optimization tool to be used for such problems need to be chosen carefully and correctly.

The articulate or jointed arm robot (or sometime called Anthropomorphic arms) closely resembles the human arm. The mechanical structure has at least three rotary joints which forms a polar coordinate system. The basic three rotary joints able Arm swap, shoulder swivel and elbow rotations. Additional 3 revolute joints (Roll, Yaw, and Pitch) and one prismatic joint allow the robot to point in many directions, and then reach out some radial distance.

This structure is very flexible and has the ability to reach over obstructions. It can generally achieve any position and orientation within the working envelope. As such articulate robots are used for a wide range of applications including paint spraying, arc and spot welding, machine tending, etc. For examples, the articulate robot allows the welding torch to be manipulated in almost the same fashion as a human being would manipulate it. The torch angle and travel angle can be changed to make good quality welds in all positions. Articulate robots also allow the arc to weld in areas that are difficult to reach. In addition, articulate robots are compact and provide the largest work envelope relative to their size. Typical articulate robots have five or six free programmable arms or axes. As mentioned, the flexibility of the articulate robots makes them well suit for a wide variety of industrial application. But, it is not easy to control. When driving these robots in their natural co-ordinate system (joint space) the motion of the robot from one point to another can be difficult to visualize as the robot will move each joint through the minimum angle required. This means that the motion of the tool will not be a straight line.

### 4.1.1 Robot specification

In order to select a robot for a specific application one must look at some important specification of robot. Some of the pertinent parameters of an industrial robot specification are as follows.

Accuracy: When the robot's program instruct the robot to move to a specified point, it does not actually perform as per specified. The accuracy measure such variance that is, the distance between the specified position that a robot is trying to achieve (programming point), and the actual X, Y and Z resultant position of the robot end effector.

Repeatability: The ability of a robot to return repeatedly to a given position. It is the ability of a robotic system or mechanism to repeat the same motion or achieve the same position.

Degree of Freedom (DOF): Each joint or axis on the robot introduces a degree of freedom. Each DOF can be a slider, rotary, or other type of actuator. The number of DOF that a manipulator possesses thus is the number of independent ways in which a robot arm can move.

Resolution: The smallest increment of motion or distance that can be detected or controlled by the robotic control system. It is a function of encoder pulses per revolution and drive (e.g. reduction gear) ratio.

Envelope: A three-dimensional shape that defines the boundaries that the robot manipulator can reach; also known as reach envelope.

- Maximum envelope: the envelope that encompasses the maximum designed movements of all robot parts, including the end effector, workpiece and attachments.
- Restricted envelope is that portion of the maximum envelope which a robot is restricted by limiting devices.
- Operating envelope: the restricted envelope that is used by the robot while performing its programmed motions.

Reach: The maximum horizontal distance from the center of the robot base to the end of its wrist.

Maximum speed: A robot moving at full extension with all joints moving simultaneously in complimentary directions at full speed. The maximum speed is the theoretical values which does not consider under loading condition.

Payload**:** The maximum payload is the amount of weight carried by the robot manipulator at reduced speed while maintaining rated precision.

Three groups of attributes, as shown in Table 4.1, are used to evaluate robots.The attributes can be grouped under engineering attributes, vendor-related attributes and cost attributes.

1. Engineering Attributes: determine the ability of robots to perform tasks and include load capacity, speed, repeatability, and accuracy.
2. Vendor-related Attributes: determine the attractiveness of robot vendors.
3. Cost Attributes: determine total costs of installing and operating robots.

Table 4.1 Attributes used for evaluating robots

| Engineering Attributes | Vendor-Related Attributes | Cost Attributes |
|---|---|---|
| • Load Capacity<br>• Repeatability<br>• Velocity<br>• Programming Method<br>• Vertical Reach<br>• Horizontal reach<br>• Memory size<br>• Accelaration<br>• Deceleration<br>• Degrees of freedom<br>• Reliability<br>• Diagnostic capability<br>• Programming Language<br>• Software<br>• Control type<br>• Recovery from error | • Brand<br>• Availability of Training<br>• Quality of Training<br>• Documentation<br>• Installation Support<br>• Spare parts avalibility<br>• Installation leadtime<br>• Pre-sale services<br>• Servicing ability<br>• Warranty | • Robot Cost<br>• Installation Cost<br>• Tooling Cost<br>• Energy Consumptiom<br>• Labour Cost<br>• Maintenance cost |

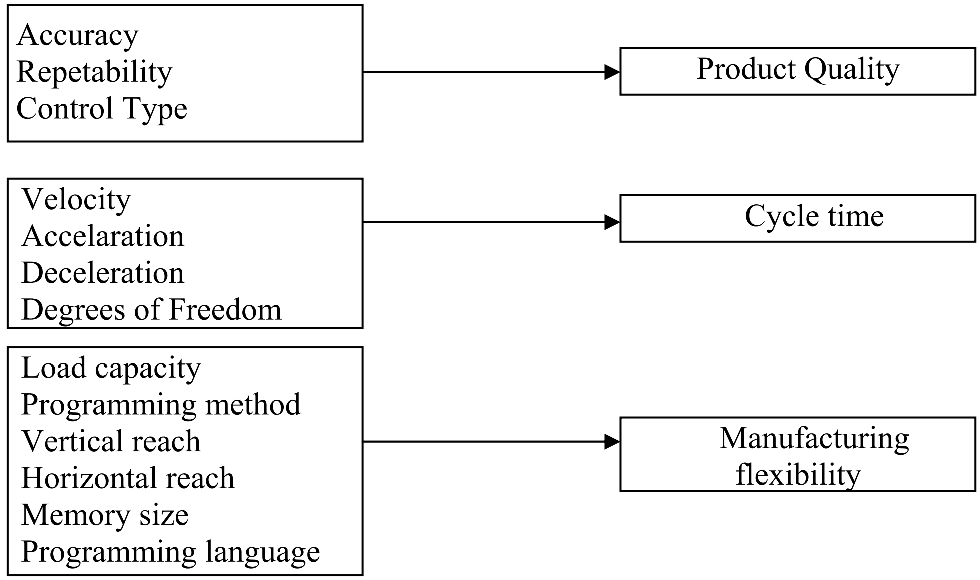



Figure 4.1 Robot engineering attributes and performance of production systems

Efforts need to be extended to determine attributes that influence robot selection for a given industrial application, using a logical approach to eliminate unsuitable robots, and for selection of a proper robot to strengthen the existing robot selection procedure. Pertinent attributes and the alternative robots involved are to be identified. Values of the attributes and their relative importance are to be obtained. An objective or subjective value, or its range, may be assigned to each identified attribute as a limiting value, or threshold value, for its acceptance for the considered robot selection problem. An alternative robot with each of its selection attributes, meeting the acceptance value, may be short-listed. After short-listing the alternative robots, the main task to choose the alternative robot is to see how it serves the attributes considered. Engineering attributes, as shown in Figure 4.1, have a critical effect on performance of production systems.

### 4.2 Manipulator attributes

Proper identification of manipulator attributes is critically important when comparing various alternative robots. Whenever a robot user desires to purchase or select a new robot, this identification of attributes attain significant importance. However, in most cases the user needs to be assisted in identifying the robot attributes logically. For the purpose of ranking, a robot manipulator can be specified by a number of quantitative attributes such as payload capacity, repeatability, horizontal reach, etc. However all attributes cannot be expressed quantitatively. Such attributes may be built quality, after sales service etc. These attributes may be expressed by a rate on the scale (say 1-10). There are some attributes which are informative in nature, such as type of drive (electrical, pneumatic etc.), the coordinate system (polar, cylindrical, rectangular etc.), which may be denoted by some number whose numerical value will have no significance .There are some attributes for which the quantification is not available. For instance, reliability can be expressed in terms of Mean Time Between Failure (MTBF) or Mean Time to Repair (MTTR) methods. Attributes like life expectancy may be estimated through experimentation, if not mentioned by the manufacturer. The identification of various pertinent attributes and their values, rates and estimates help the user for create a database for storage and retrieval which can be used in different formats for different purposes by different people.

Table 4.2 Manipulator attributes

| Attribute type | Parameter |
|---|---|
| General | Price range, Type of robot and Coordinate system |
| Physical | Type of actuators, Weight of the robot, Size of the robot, Type of grippers supported, Number of axes, Space requirements of the robot |
| Performance | Payload of the robot, Workspace, Stroke, Maximum end effector speed, Accuracy, Repeatability, Resolution |
| Structure/architecture | Degree of freedom ,Type of joints |
| Application | Working environment |
| Sophistication | Maintainability and Safety features |
| Control/feedback system | Control of robotic joints, Gripper control, Sensors, Programming method, Number of input and output channels of the controller |
| Availability/reliability | Downtime and Reliability |

These data individually or collectively help the user to select the most suitable robot for a task that he intends to perform. The computational simplicity of the equations of motion is also an index of performance characteristics of robot. This simplicity has been identified as an attribute. The structure of the manipulator is very important feature of the manipulator and also affects the performance. These attributes get their fair representation in formulating the present model. The robot operating in its workspace does not operate with same ease everywhere. This ease of operation, termed as manipulability, can be quantified as manipulability measure and can be used as an attribute. The motion provided by actuators and motion gained with the basic structure of robot, i.e., motion transformation is also an important robot characteristics. Appropriate quantification methods are required to be standardized to guide the manufacturers for quantifying the attributes. The main attributes have been broken down to sub-attributes and sub-sub-attributes so that the robot manipulator can be identified in very precise and detailed manner. The manipulator attributes based on its broad area as general parameters, physical parameters, performance based, etc. are given in Table 4.2.

### 4.3 The robot selection process

The attributes can be coded as per the parameters coding scheme in Table 4.3. The information supplied by the manufacturer to the user is meager and it is required to be more elaborate. The '0' represents that the information relating to the particular cell is not available, but it should be provided to make the database exhaustive. This coding scheme can be used as it is for the visual comparison between two robots up to certain extent. It allows faster comparison in various formats. The identification code in Table 4.4 specifies the attribute information with the allotted code in the respective cells.

Table 4.3 Parameter coding scheme

| Parameter | Parameter coding scheme | | | | | | | Code | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| General | 1 | 2 | 3 | | | | | 0 | 9 | | 4 | | | |
| Physical | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 0 | 0 | 13 | 0 | 0 | 0 | 0 |
| Performance | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 5 | 0 | 0 | 2 | 0 | 5 | 0 |
| Structure | 18 | 19 | | | | | | 1 | 0 | | | | | |
| Application | 20 | | | | | | | 0 | | | | | | |
| Sophistication | 21 | 22 | | | | | | 0 | 0 | | | | | |
| Control | 23 | 24 | 25 | 26 | 27 | 28 | | 0 | 0 | 0 | 1 | 0 | 0 | |
| Availability | 29 | 30 | | | | | | 0 | 0 | | | | | |

Table 4.4 Identification code

| Sl. | Attribute | Information | Code |
|---|---|---|---|
| 1. | Price range | $ 19500 | 9 |
| 2. | Type of robot | | 0 |
| 3. | Swept area | 50 deg/sec | 4 |
| 4. | Type of actuators | | 0 |
| 5. | Weight of the robot | | 0 |
| 6. | Reach | 800mm | 13 |
| 7. | Type of grippers supported | | 0 |
| 8. | Number of axes | | 0 |
| 9. | Space requirements of the | | 0 |
| 10. | Types of end effectors | | 0 |
| 11. | Payload of the robot | 4kg | 5 |
| 12. | Workspace | | 0 |
| 13. | Stroke | | 0 |
| 14. | Maximum end effector speed | 1.0 m/sec | 2 |
| 15. | Accuracy | | 0 |
| 16. | Repeatability | ±0.1mm | 5 |
| 17. | Resolution | | 0 |
| 18. | Degree of freedom | 2 | 1 |
| 19. | Type of joints | | 0 |
| 20. | Working environment | | 0 |
| 21. | Maintainability | | 0 |
| 22. | Safety features | - | 0 |
| 23. | Control of robotic joints | | 0 |
| 24. | Gripper control | - | 0 |
| 25. | Sensors | | 0 |
| 26. | Programming method | | 0 |
| 27. | Number of input channels | - | 0 |
| 28. | Number of output channels | - | 0 |
| 29. | Down time | | 0 |
| 30. | Reliability | | 0 |

In general, the robot selection criteria include some key specifications such as; degrees of freedom, pay load, swept area, maximum reach, maximum speed, cost, and repeatability. Though most of the attributes have been identified, all of them may not be important for the intended application. There will be few attributes, which will have direct effect on the selection procedure. Some of the attributes may be selected as 'pertinent attributes' as necessitated by the particular application and/or the user. The threshold values to these 'pertinent attributes' may be assigned by obtaining information from the user and the group of experts. The selection procedure focuses solely on the pertinent attributes leaving out the rest. On the basis of the threshold values, a shortlist of robots is obtained. This is achieved by scanning the database for those attributes, one at a time and eliminating the robot alternatives, which have one or more of these attribute values that fall short of threshold values. The step-wise activities for ranking and selecting the robots are presented by a flowchart (Figure 4.2). The robot selection architecture system is divided into four activities; i)operation requirements and data library of robots, ii) coding scheme, iii) selection of attributes, and iv) ranking of robots.

The first step here will be to represent all the information available from the database about these satisfying solutions in the matrix form. Such a matrix is called as decision matrix, D. Each row of this matrix is allocated to one candidate robot and each column to one attribute under consideration. Therefore an element $d_{ij}$ of the decision matrix D gives the value of $j_{th}$ attribute in the row (non-normalized) form and units, for the $i_{th}$ robot. Thus if the number of short-listed robots is 'm' and the number of pertinent attributes is 'n', the decision matrix is an m × n matrix.

Normalization is used to bring the data within particular range that provides the dimensionless magnitudes. The normalized specification matrix has the magnitudes of all the attributes of the robots on the common scale of 0 to 1. An element $n_{ij}$ of the normalized matrix 'N' can be calculated as;

$$n_{ij} = \frac{d_{ij}}{\left(\sum_{i=1}^{m} d_{ij}^{2}\right)^{1/2}} \tag{4.1}$$

Where, $d_{ij}$ is an element of the decision matrix, 'D'. Information is gathered in terms of a ratio from the user or the experts on the relative importance of one attribute with respect to another. All such pair-wise comparisons are stored in a matrix called relative importance matrix, 'A'. Here, $a_{ij}$ contains the relative importance of $i^{th}$ attribute over the $j^{th}$ attribute. A mini-database is thus formed which comprises of these satisfying solutions. The problem is now one of finding out the optimum or best out of these satisfying solutions.

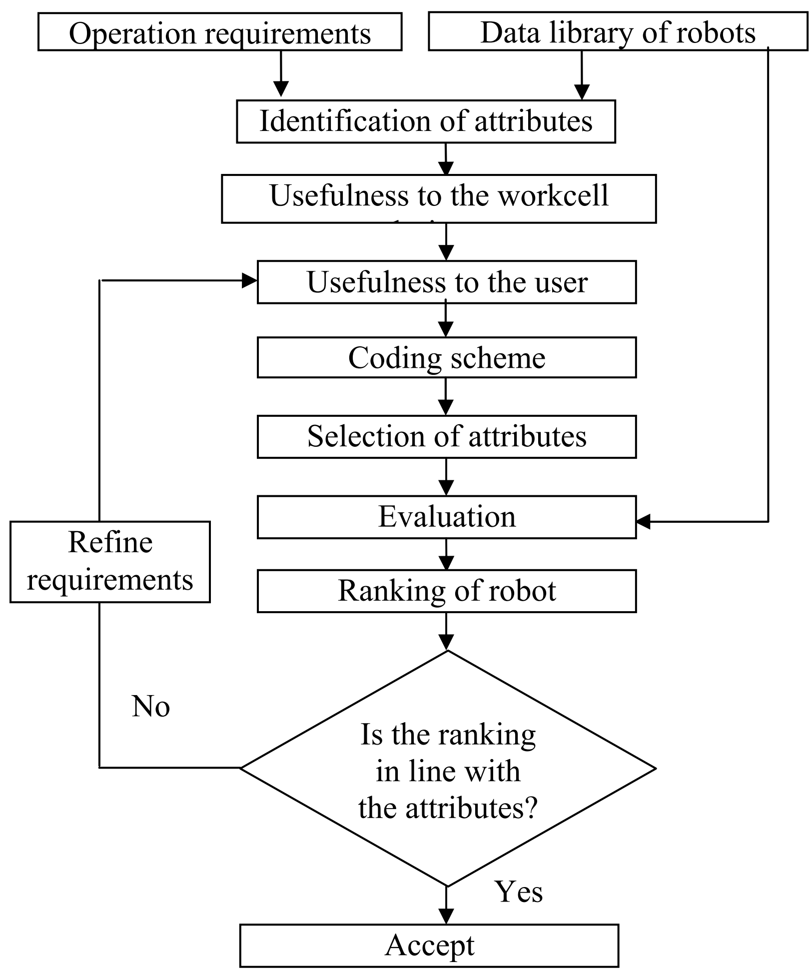



Figure 4.2 Robot selection procedures

The first step is to represent all the information available from the database about these satisfying solutions in the matrix form called as decision matrix, 'D'. Each row of this matrix is allocated to one candidate robot and each column to one attribute under consideration. An element dij of the decision matrix 'D' gives the value of jth attribute in the row (non-normalized) form and units, for the ith robot. The next step is construction of the normalized specification matrix, 'N', from the decision matrix, 'D'.

The symmetric terms of this matrix are reciprocals of each other while the diagonal is unity. This matrix is then modified into a representation that gives the relative weights of all attributes taken together so that the cumulative sum of the weights is equal to unity. The eigen vector method is used to find the weights. The eigen vector method seeks to find weight vector 'w' from the eigen value of the matrix, 'A'. The weighted normalized matrix 'V' combines the relative weights and normalized specification of the candidates gives the true comparable values of the attributes. Therefore,

$$V=\begin{bmatrix} w_1n_{1,1} & w_2n_{1,2} & \cdots & w_nn_{1,n} \\ w_1n_{2,1} & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ w_1n_{m,1} & w_2n_{m,2} & \cdots & w_nn_{m,n} \end{bmatrix}=\begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,n} \\ v_{2,1} & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ v_{1,1} & v_{1,1} & \cdots & v_{1,1} \end{bmatrix} \tag{4.2}$$

### 4.3.1 Ranking and selection procedure

The weighted normalized matrix V is used to obtain the +ve and -ve benchmark robots. The benchmark robots are hypothetical robots, which are supposed to have the best and the worst possible attribute magnitudes. The method is based on the concept that the chosen option (optimum) has the shortest distance from the +ve benchmark robot and is farthest from the -ve benchmark robot. This measure ensures that the top ranked robot is closest to +ve benchmark robot and is farthest from -ve benchmark robot. The separation measures are calculated from +ve and -ve benchmark robots, respectively, as $S_i^*$ and $S_i^-$. These are given by

$$S_i^* = \left[ \sum_{j=1}^{n} (v_{ij} - v_1^*)^2 \right]^{1/2} \quad (i=1,2,\ldots,m) \tag{4.3}$$

and $S_i^- = \left[ \sum_{j=1}^{n} (v_{ij} - v_1^-)^2 \right]^{1/2}$ ($i$=1,2,…,n) (4.4)

Then the relative closeness to the +ve benchmark robot, C*, which is a measure of the suitability of the robot can be calculated using equation (5).

$$C^* = S_i^- / (S_i^* + S_i^-) \quad (4.5)$$

A robot with the largest C* is preferable. Ranking of the candidate robots in accordance with the decreasing values of C* is done.

## 4.4 Illustrative examples

Now, to select the robots and validate the application of attribute based methods, three examples are considered. A pick-and-place task is considered with a suitable robot. The minimum requirement for this application is tabulated as shown in Table 4.5. After 'elimination search' using the generated database, a shortlist of candidate robots and their pertinent attributes are prepared as given in Table 4.6.

Table 4.5 Minimum requirement of a robot

| Sl | Parameter | Values |
|---|---|---|
| 1 | Load capacity | minimum 4 kg |
| 2 | Repeatability | 0.5 mm |
| 3 | Speed | at least 800 mm/s |
| 4 | Types of drives | (Actuators) electrical only |
| 5 | Reach | 500 mm |
| 6 | Degree of freedom | at least 4 |
| 7 | Swept area | $270^0$ |

Table 4.6 Fixed costs and parameter values the short-listed robots

| | Robot-1 | Robot-2 | Robot-3 | Robot-4 |
|---|---|---|---|---|
| Specification | Puma (560-c) | (Adept one XL) | Fanuc Arcmate Sr.R.J | Staubli RX 130B |
| DOF | 6 | 4 | 6 | 6 |
| Pay Load | 4 kg | 12 kg | 10 kg | 12 kg |
| Swept Area | $320^{\circ}$ | $270^{\circ}$ | $300^{\circ}$ | $320^{\circ}$ |
| Max. Reach | 878 mm | 800 mm | 1529 mm | 1250 mm |
| Max Speed | 1.0 m/sec | 1.2 m/sec | 3.60 m/sec | 3.09m /sec |
| Cost | $35,000 | $19,500 | $56,400 | $60,000 |
| Repeatability | 0.1 mm | 0.025 mm | 0.1 mm | 0.03 mm |

### 4.4.1 Selection of robots on the basis of fitness

An example is considered to validate the application of the attribute based selection process. This example problem considers four robots with seven attributes. Since repeatability has the smallest magnitude amongst the attributes, the reciprocal of its values are taken in the decision matrix ‘D’ along with the actual values of other attributes. The procedure for the selection of the robot is as follows:

Step 1: Formation of decision matrix, ‘D’.

$$D = \begin{bmatrix} \text{DOF} & \text{Payload} & \text{Swept Area} & \text{Reach} & \text{Speed} & \text{Cost} & \text{Repeatability} \\ 6 & 4 & 320 & 878 & 1.0 & 35{,}000 & 10 \\ 4 & 12 & 270 & 800 & 1.2 & 19{,}500 & 40 \\ 6 & 10 & 300 & 1529 & 3.6 & 56{,}400 & 10 \\ 6 & 12 & 320 & 1250 & 3.09 & 60{,}000 & 33.3 \end{bmatrix}$$

Step 2: Construction of relative importance matrix 'A'.

$$A = \begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 3 & 0.5 \\ 1 & 1 & 1 & 1 & 1 & 2 & 0.33 \\ 0.5 & 1 & 1 & 2 & 1 & 2 & 0.33 \\ 0.5 & 1 & 0.5 & 1 & 0.5 & 1.5 & 0.2 \\ 0.5 & 1 & 1 & 2 & 1 & 1.5 & 0.2 \\ 0.33 & 0.5 & 0.5 & 0.666 & 0.666 & 1 & 0.2 \\ 2 & 3 & 3 & 5 & 5 & 5 & 1 \end{bmatrix}$$

Step 3: Calculation of maximum eigen value of 'A'.

where $\lambda$ is the eigen value of A

$$(A - \lambda_{max} I) = \begin{bmatrix} -6.1314 & 1 & 2 & 2 & 2 & 3 & 0.5 \\ 1 & -6.1314 & 1 & 1 & 1 & 2 & 0.33 \\ 0.5 & 1 & -6.1314 & 2 & 1 & 2 & 0.33 \\ 0.5 & 1 & 0.5 & -6.1314 & 0.5 & 1.5 & 0.2 \\ 0.5 & 1 & 1 & 2 & -6.1314 & 1.5 & 0.2 \\ 0.33 & 0.5 & 0.5 & 0.666 & 0.666 & -6.1314 & 0.2 \\ 2 & 3 & 3 & 5 & 5 & 5 & -6.1314 \end{bmatrix}$$

Since $(A-\lambda_{max} I) = 0$, and $\lambda$= 7.1314, 0.0066 $\pm$ 0.8966 i, 0.0075$\pm$ 0.4081i-0.0798 $\pm$ 0.0356 i, $\lambda_{max}$= 7.1314.

Step 4: Calculation of weights: Since $(A- \lambda_{max}I)\ w = 0$ and $(w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7) = 1$; the weights are found to be as follows.

$w_1 = 0.1724$, $w_2 = 0.1145$, $w_3 = 0.1132$, $w_4 = 0.0766$, $w_5 = 0.1024$, $w_6 = 0.0591$, and $w_7 = 0.3618$.

Step 5: Calculation of the normalized specification matrix.

$$N = \begin{bmatrix} 0.436 & 0.852 & 0.485 & 0.572 & 0.73 & 0.45 & 0.682 \\ 0.655 & 0.284 & 0.572 & 0.628 & 0.608 & 0.807 & 0.17 \\ 0.436 & 0.34 & 0.454 & 0.328 & 0.202 & 0.278 & 0.682 \\ 0.436 & 0.284 & 0.485 & 0.417 & 0.235 & 0.263 & 0.204 \end{bmatrix}$$

Step 6: Calculation of the weighted normalized specification matrix.

$$V = \begin{bmatrix} 0.075 & 0.097 & 0.054 & 0.043 & 0.074 & 0.026 & 0.246 \\ 0.1129 & 0.032 & 0.064 & 0.048 & 0.062 & 0.047 & 0.061 \\ 0.075 & 0.038 & 0.051 & 0.025 & 0.02 & 0.016 & 0.246 \\ 0.075 & 0.032 & 0.054 & 0.031 & 0.024 & 0.015 & 0.073 \end{bmatrix}$$

The weighted normalized attributes for the +ve and –ve benchmark robots can be obtained as follows.

$V^{*}$= (0.1129, 0.097, 0.064, 0.048, 0.074, 0.047, 0.246)

$V^{-}$ = (0.075, 0.032, 0.051, 0.025, 0.02, 0.015, 0.061)

The separation from the +ve and –ve benchmark robots are found as;

$$S_1^* = 0.044, S_2^* = 0.195, S_3^* = 0.096, S_4^* = 0.196$$

$$S_1^- = 0.206, S_2^- = 0.069, S_3^- = 0.185, S_4^- = 0.0083$$

### 4.4.2 Selection of robots on the basis of capability

The example problem considers four comparable robots (viz. R-1, R-2, R-3 and R-4) and their pertinent parameters are listed out. They are tabulated in Table 4.7. Out of all the parameters listed, repeatability has the smallest magnitude. Hence the reciprocal of the repeatability values are taken in the decision matrix ‘D’ along with the actual values of other parameters for further calculations.

The procedure for the selection of the robot is as follows:

Step 1: Formation of decision matrix, 'D'.

$$D = \begin{bmatrix} \text{DOF} & \text{Payload} & \text{Swept Area} & \text{Reach} & \text{Speed} & \text{Cost} & \text{Repeatability} \\ 6 & 4 & 320 & 878 & 1.0 & 35{,}000 & 10 \\ 4 & 12 & 270 & 800 & 1.2 & 19{,}500 & 40 \\ 6 & 10 & 300 & 1529 & 3.6 & 56{,}400 & 10 \\ 6 & 12 & 320 & 1250 & 3.09 & 60{,}000 & 33.3 \end{bmatrix}$$

Step 2: Formation of weight matrix, 'W'.

$$W = \begin{bmatrix} \text{Speed} & \text{Re ach} & \text{Swept Area} & \text{DOF} & \text{Payload} & \text{Cost} & \text{Re peatability} \\ 2 & 13 & 4 & 1 & 6 & 9 & 0.5 \\ 3 & 12 & 5 & 2 & 5 & 8 & 1 \\ 4 & 11 & 6 & 3 & 4 & 7 & 1.5 \\ 5 & 10 & 7 & 4 & 3 & 6 & 2 \\ 6 & 9 & 8 & 5 & 2 & 5 & 2.5 \\ 7 & 8 & 9 & 6 & 1 & 4 & 3 \\ 8 & 7 & 9 & 6 & 1 & 4 & 3 \\ 9 & 6 & 8 & 5 & 2 & 5 & 2.5 \\ 10 & 5 & 7 & 4 & 3 & 6 & 2 \\ 11 & 4 & 6 & 3 & 4 & 7 & 1.5 \\ 12 & 3 & 5 & 2 & 5 & 8 & 1 \\ 13 & 2 & 4 & 1 & 6 & 9 & 0.5 \end{bmatrix}$$

Step 3: Calculation of the normalized specification matrix.

$$N = \begin{bmatrix} 0.436 & 0.852 & 0.485 & 0.572 & 0.73 & 0.45 & 0.682 \\ 0.655 & 0.284 & 0.572 & 0.628 & 0.608 & 0.807 & 0.17 \\ 0.436 & 0.34 & 0.454 & 0.328 & 0.202 & 0.278 & 0.682 \\ 0.436 & 0.284 & 0.485 & 0.417 & 0.235 & 0.263 & 0.204 \end{bmatrix}$$

Step 4: Calculation of normalized value (N.V.), ranking factor ($\sigma = W * N$), and total score $\sum \sigma$.

Step 5: Calculating the average of the ranking factors of all the robots.

Table 4.7 Ranking factor with one set of weightage

| | Parameter | Value | Normalized value | Weight | Ranking factor($\sigma$) | $\sum\sigma$ |
|---|---|---|---|---|---|---|
| R-1 | Speed | 1000 | 0.075 | 2 | 0.15 | 2.471 |
| | Max.Reach | 878 | 0.097 | 13 | 1.261 | |
| | Swept area | 5.58 | 0.054 | 4 | 0.216 | |
| | DOF | 6 | 0.043 | 1 | 0.043 | |
| | Payload | 4 | 0.074 | 6 | 0.444 | |
| | Cost | 35,000 | 0.026 | 9 | 0.234 | |
| | Repeatability | 0.1 | 0.246 | 0.5 | 0.123 | |
| R-2 | Speed | 1200 | 0.112 | 2 | 0.2258 | 1.771 |
| | Max.Reach | 800 | 0.032 | 13 | 0.416 | |
| | Swept area | 4.712 | 0.064 | 4 | 0.256 | |
| | DOF | 4 | 0.048 | 1 | 0.048 | |
| | Payload | 12 | 0.062 | 6 | 0.372 | |
| | Cost | 19,500 | 0.047 | 9 | 0.423 | |
| | Repeatability | 0.025 | 0.061 | 0.5 | 0.0305 | |
| R-3 | Speed | 3600 | 0.075 | 2 | 0.15 | 1.26 |
| | Max.Reach | 1529 | 0.038 | 13 | 0.494 | |
| | Swept area | 5.235 | 0.051 | 4 | 0.204 | |
| | DOF | 6 | 0.025 | 1 | 0.025 | |
| | Payload | 10 | 0.02 | 6 | 0.12 | |
| | Cost | 56,400 | 0.016 | 9 | 0.144 | |
| | Repeatability | 0.1 | 0.246 | 0.5 | 0.123 | |
| R-4 | Speed | 3090 | 0.075 | 2 | 0.15 | 1.285 |
| | Max.Reach | 1250 | 0.032 | 13 | 0.416 | |
| | Swept area | 5.585 | 0.054 | 4 | 0.216 | |
| | DOF | 6 | 0.031 | 1 | 0.031 | |
| | Payload | 12 | 0.024 | 6 | 0.144 | |
| | Cost | 60,000 | 0.015 | 9 | 0.135 | |
| | Repeatability | 0.03 | 0.073 | 0.5 | 0.0365 | |

### 4.4.3 Selection of robots on the basis of task requirement

In order to demonstrate and validate the methodology of the proposed method five robots with different configurations and capabilities are considered. The objective values of the robot selection attributes, which are given in Table 4.8.

Table 4.8 Criteria for robot selection

| Criteria | Robot-1 | Robot-2 | Robot-3 | Robot-4 | Robot-5 |
|---|---|---|---|---|---|
| Maximum Reach(MR) | 1000 | 2000 | 5000 | 5000 | 5500 |
| DOF(DF) | 2 | 3 | 4 | 5 | 6 |
| Payload(PL) | 5 | 10 | 30 | 40 | 60 |
| Velocity(VL) | 50 | 90 | 120 | 200 | 250 |
| Arm geometry(AG) | 4 | 9 | 20 | 20 | 24 |
| Actuator(AT) | 7 | 10 | 3 | 10 | 7 |
| Control mode(CM) | 4 | 6 | 8 | 10 | 8 |
| Repeatability(RT) | 0.02 | 0.1 | 0.5 | 1.0 | 1.0 |
| Robot programming(RP) | 3 | 4 | 6 | 6 | 8 |
| Space(SC)* | 0.5 | 0.45 | 0.4 | 0.3 | 0.2 |
| Time(TE)* | 0.359 | 0.3 | 0.28 | 0.3 | 0.2 |
| DOF(DF1)* | 3 | 3 | 3 | 3 | 3 |
| Force (FR)* | 5 | 5 | 5 | 5 | 5 |

*These values pertain to task-1 of the fifteen tasks actually considered for the problem. However only one task has been considered for calculation, similarly other calculations are to be made. The normalized values of all these parameters are taken to form the decision matrix.

The procedure for the selection of the robot is as follows:

Step 1: Formation of decision matrix, ‘D’.

$$D = \begin{array}{c} \begin{array}{ccccccccccccc} MR & DF & PL & VL & AG & AT & CM & RT & RP & SC & TE & DF1 & FR \end{array} \\ \begin{bmatrix} 1000 & 2 & 5 & 50 & 4 & 7 & 4 & 50 & 3 & 2 & 2.78 & 3 & 5 \\ 2000 & 3 & 10 & 90 & 9 & 10 & 6 & 10 & 4 & 2.22 & 3.33 & 3 & 5 \\ 5000 & 4 & 30 & 120 & 20 & 3 & 8 & 2 & 6 & 2.5 & 3.57 & 3 & 5 \\ 5000 & 5 & 40 & 200 & 20 & 10 & 10 & 1 & 6 & 3.33 & 3.33 & 3 & 5 \\ 5500 & 6 & 60 & 250 & 24 & 7 & 8 & 1 & 8 & 5 & 5 & 3 & 5 \end{bmatrix} \end{array}$$

Step 2: Formation of weight matrix, ‘W’.

$$
W = \begin{array}{c}
\begin{array}{ccccccccccccc}
MR & DF & PL & VL & AG & AT & CM & RT & RP & SC & TE & DF1 & FR
\end{array} \\
\begin{bmatrix}
13 & 2 & 6 & 1 & 3 & 1 & 6 & 0.5 & 1 & 1 & 6 & 13 & 2 \\
12 & 3 & 5 & 2 & 4 & 2 & 5 & 1 & 1.5 & 2 & 5 & 12 & 3 \\
11 & 4 & 4 & 3 & 5 & 3 & 4 & 1.5 & 2 & 3 & 4 & 11 & 4 \\
10 & 5 & 3 & 4 & 6 & 4 & 3 & 2 & 2.5 & 4 & 3 & 10 & 5 \\
9 & 6 & 2 & 5 & 7 & 5 & 2 & 2.5 & 3 & 5 & 2 & 9 & 6 \\
8 & 7 & 1 & 6 & 8 & 6 & 1 & 3 & 3.5 & 6 & 1 & 8 & 7 \\
7 & 8 & 1 & 6 & 9 & 6 & 1 & 3 & 3.5 & 6 & 1 & 7 & 8 \\
6 & 9 & 2 & 5 & 10 & 5 & 2 & 2.5 & 3 & 5 & 2 & 6 & 9 \\
5 & 10 & 3 & 4 & 11 & 4 & 3 & 2 & 2.5 & 4 & 3 & 5 & 10 \\
4 & 11 & 4 & 3 & 12 & 3 & 4 & 1.5 & 2 & 3 & 4 & 4 & 11 \\
3 & 12 & 5 & 2 & 13 & 2 & 5 & 1 & 1.5 & 2 & 5 & 3 & 12 \\
2 & 13 & 6 & 1 & 14 & 1 & 6 & 0.5 & 1 & 1 & 6 & 2 & 13
\end{bmatrix}
\end{array}
$$

Step 3: Calculation of the normalized specification matrix.

$$
N = \begin{bmatrix}
0.844 & 0.713 & 0.877 & 0.796 & 0.876 & 0.344 & 0.691 & 0.979 & 0.674 & 0.259 & 0.339 & 0.447 & 0.447 \\
0.428 & 0.475 & 0.438 & 0.437 & 0.388 & 0.241 & 0.461 & 0.195 & 0.505 & 0.287 & 0.398 & 0.447 & 0.447 \\
0.171 & 0.356 & 00.146 & 0.331 & 0.174 & 0.803 & 0.345 & 0.039 & 0.337 & 0.323 & 0.435 & 0.447 & 0.447 \\
0.171 & 0.285 & 0.109 & 0.199 & 0.174 & 0.241 & 0.276 & 0.019 & 0.337 & 0.571 & 0.406 & 0.447 & 0.447 \\
0.155 & 0.237 & 0.072 & 0.159 & 0.145 & 0.344 & 0.345 & 0.019 & 0.252 & 0.647 & 0.61 & 0.447 & 0.447
\end{bmatrix}
$$

Step 4: Calculation of normalized value (N.V.), ranking factor ($\sigma$ =W * N), and total score $\sum\sigma$.

Step 5: Determination of the average of the ranking factors of all the robots.

Table 4.9 Ranking factor with one set of weightage of robot-1

| Parameter | Value | Normalized value | W | Ranking factor(σ) | ∑ σ |
|---|---|---|---|---|---|
| MR | 1000 | 0.844 | 13 | 10.972 | 35.7715 |
| DF | 2 | 0.713 | 2 | 1.462 | |
| PL | 5 | 0.877 | 6 | 5.262 | |
| VL | 50 | 0.796 | 1 | 0.796 | |
| AG | 4 | 0.876 | 3 | 2.628 | |
| AT | 7 | 0.344 | 1 | 0.344 | |
| CM | 4 | 0.691 | 6 | 4.146 | |
| RT | 0.02 | 0.979 | 0.5 | 0.4895 | |
| RP | 3 | 0.674 | 1 | 0.674 | |
| SC | 0.5 | 0.259 | 1 | 0.259 | |
| TE | 0.359 | 0.339 | 6 | 2.034 | |
| DF1 | 3 | 0.447 | 13 | 5.811 | |
| FR | 5 | 0.447 | 2 | 0.894 | |

The calculation of the total ranking factor, ∑ σ, for one set weights in robot-1 is presented in Table 4.9. The calculations of ranking factors are made for the other robots with a total of 12 different sets of weights.

### 4.4.4 Selection of robots on the basis of case based approach

The performance of industrial robots is often specified using many parameters are the important and practical parameters [85]. Repeatability, accuracy, load capacity, and velocity. Industry-wide standards for measuring these parameters are not yet fully established; however, improved methods for analysing robot performance are being developed [86]. A large number of attributes for robot selection, and ranked the robots using TOPSIS and graphical methods [87], comparing the rankings given by these methods.However, the weights assigned by the authors to the attributes were not consistent. Khouja and Kumar [88] used options theory and an investment evaluation procedure for selection of robots. An investment evaluation using data envelopment analysis for robot selection is carried out [89]. A decision support system [90] based on analytical algorithms to select machining centers and robots concurrently from the market milien.

It is a fact that practical solutions are better obtained when theoritical modelling of the problem is reinforced with experience.The longer the experience, the better.The present work aims at developing a methodology for selection of robots for industrial applications where the knowledge about the system, the environment, old cases of similar nature are considered apart from the robot's data and task requirement.

### 4.4.5 The process structure and its components

In order to make the robot application process efficient, the selection of robots must be approached in an effective and systamatic methodology. Implementing a robotic application system is best done in the process that involves not only the robot, but also the tasks, the entire system, and the environment. The proposed robot selection procedure follows the main route according to the following five major activities:

1. Information database - develop main applications, system performance, system requirements, and justification.
2. Indexing - identify features, match using similarity case.
3. Initial solution - choose the most similar and feasible robot application cases.
4. Iteration - modify solutions to fit the current robot selection query.
5. Implementation - apply workcell design, cost estimation, design review.

These activities are briefly explained as follows.

*A. Information database*

The database contains useful information pertaining to the robots and the tasks, system and its environments under consideration.

a) The robots: The following information regarding the candidate robots need to be explored and recorded.

i) Specification: Proper identification of manipulator attributes is critically important when comparing various alternative robots. Therefore, whenever a robot user goes to the supplier for purchase of new robot, or looks at the existing robots, the identification of attributes attain significant importance. The robots may have large number features to offer.But for the purpose of the present work, only those features which, make modelling the task performing capability are picked up.

Motion requirement:Optimum speed with which the task need to be handled.Too slow a speed will result in higher cycle time ,while too large speed may not be permissible for high precision work.

Handling requirement: This is essentially looked at to determine the number of orientational /positional changes required while handling the task.This otherwise, means the robot must have sufficient DOF.

Grasping requirement: It is necessary to know the type of grasping required which include the size, (dimensions), shape, volume, weight apart from the physical properties (such as, hard, soft, solid, liquid, hot, normal etc.). Additionally, things like the status of object e.g uncovered, covered, as-is, contained etc. should also be recorded.

ii) Major attributes: Generally, there are two typologies of robot attributes: objective attributes and subjective attributes. Objective attributes are measured and defined in numerical terms. They are engineering attributes such as load capacity, accuracy, repeatability, speed, etc. or cost attributes such as purchase and installation cost, maintenance cost, training cost, etc. The subjective attributes, on the other hand (such as the vendor's service quality, the programming flexibility, the man- machine interface, etc.) are qualitative and cannot be precisely and numerically measured by the decision maker.

b) The task: Assignment of task to the robots needs a thorough analysis of the tasks before selecting an appropriate candidate robot.The task's requirement are listed down in the context of employment of robot.These requirements are the outcome of the study on the motion requirement,handling requirement holding (grasping) requirement etc.

DOF requirement: The number of joints, axes, and DOF were taken into account but the sequence of joints and their respective orientations and arrangements had not been considered.

ii) Speed requirement: In the present day robots are used for various applications, and improvement of robot performance such as high speed motion and high precision positioning is strongly required.

iii) Payload: The static load and the dynamic load during handling and positioning, or the amount of force / torque during assembly is estimated.

iv) Layout: Position, location, orientation, feeding type, rate etc. define the layout.

Tasks that are similar in terms of their requirements for robot repeatability, load capacity, reach and so on will have similar values of the membership coefficients. Computing the reach and travel distance required for each task requires information about the final cell layout which has not yet been determined. While the final layout, information will not be available until robots have been selected.

c) The system: This encompasses the entire focal area of the workcell.The various salient points of the system may be as follows.

i) Connectivity: The devices in the shop floor /cell networked/connected or stand alone, if connected, wheither the information channel is biodirectional or mono-directional (all connected, some connected, none-connected cases). When deploying robots to accomplish tasks in potentially unknown environments, one challenge to overcome is the lack of a global communication medium.

ii) Delivery requirement: The quantity and quality of the deliverables need to consider.

iii) Handling requirements: whether the component to be handled is soft, hard, fragile, packaged, etc.are considered.

d) The environment: Most task allocation problems are not static; they are dynamic decision problems that vary in time with phenomena such environmental changes and robot failures. Regardless of the method used for calculation, the robots' utility estimates will be inexact for a number of reasons, including specific hours of operation, structured/ unstructured, the type of environment and type of operation, and environmental change. These unavoidable characteristics of the multi-robot domain will necessarily limit the efficiency with which coordination can be achieved.

*B. Indexing*

This activity includes selecting of right kind of attributes from the master set of attributes, matching with the required task, Prioritizesthe requirements and selecting one (iniltialize). While looking at a specific robot, some of the attributes may be selected as 'pertinent attributes' as necessitated by the particular application and/or the user. The threshold values to these 'pertinent attributes' may be assigned by obtaining information from the user and the group of experts. The selection procedure focuses solely on the pertinent attributes leaving out the rest. On the basis of the threshold values, a shortlist of robots is obtained through picking -up the requirements of the task, incorporating the constraints of the environment, and prioritizing the requirements and short-listing the candidate robots.

*C. Initial solution*

Similarity assessment is a major part of the "experts" knowledge which is necessary for intelligent retrieval. In order to adopt a notion of similarity, the following assumptions are made:

- High similarity between the query problem and confirmed cases means high potential for solving problem.
- The similarity is based on a previous experiences and records.
- Similarity must provide a quantitative measurement.
- Similar problems have similar solution.
- A retrieved case is useful if it is similar to a query problem.

In our study, common approaches using quantitative similarity measurement for the retrieval of useful cases are made by as pick-up any one and looking its matching/similarity list with respect to the task and environment.

*D. Iteration*

A confirmed case (Case = Problem + Solution) is set of information entities, which stores the previous experience of robot selection. The set of information entities contain the robot specification and procedures of constructing a workcell design from robot selection to layout and space planning.This is achieved through the following steps,

- Pick-up next and follow the same procedure
- Continue the same procedure with all available robots

*E. Implementation*

Obviously, it is employed to reuse the experience in the context of the problem and complete or partially reuse or adapt according to the differences in the stored cases. In order to reuse the previous cases effectively, robotic adaptation rules are applied to the specification values and to the answers from the previous confirmed cases. The reuse strategy is presented as follows:

- Calculate further information from the specification
- Consider the differences between the previous cases and query problem
- Change feature values when it is necessary to obtain a good start
- Decide whether the final solution is good enough
- Refine several times to optimize the solution.

The simplest type of adaptation rule calculates a required value directly from the information that the user has already given.

- Make a comparision
- Draw a list of the robots as per their suitability

### 4.4.6 Selection of candidate robots

*A. Analysis of robot application*

A feasible robot must, at a minimum, have specifications that are equal to or better than the minimum requirements of an application. For example, a material handling robot is not feasible unless its specification on payload equals or exceeds the weight of the heaviest part it will handle. The minimum requirement for this application is tabulated as shown in Table 4.10. Note that a robot with specifications all equal to or better than the minimum requirements of the application may still fail to deliver the required performance during operation. This failure is because, as discussed earlier, the manufacturer's specifications may not hold simultaneously. Table 4.11 summarizes the prime considerations in the selection of an industrial robot.

Table 4.10 Minimum requirement of a Robot

| Sl | Parameter | Values |
|---|---|---|
| 1 | Working envelop | $\leq$ minimum 30 $m^3$ |
| 2 | Payload | $\leq$ 120 kg |
| 3 | Repeatability | $\pm$ 0.1 mm |
| 4 | Work lot size | $\geq$ 25 tasks |
| 5 | Number of different work pieces per Processes | $\leq$ 10 |
| 6 | Cycle time | $\geq$ 5 sec |

Table 4.11 Criteria for robot selection

| | Robot-1 | Robot-2 | Robot-3 | Robot-4 | Robot-5 |
|---|---|---|---|---|---|
| Reach (R) | ≤ 1 m. | 1m< R ≤2m | 2m< R ≤ 5m | R >5m | |
| DOF | 2 | 3 | 4 | 5 | 6 |
| Payload | 5 | 10 | 30 | 60 | 60 |
| Velocity (mm/s) | 250 | 500 | 1000 | 2500 | 5000 |
| Arm Geometry | RE | CY | SP | AR | SP |
| Actuator Types | H | EL | PN | PN | PN |
| Control Modes | NS | PTP | CP | PTP and CP | CP |
| Repeatability(MM) | 0.02 | 0.1 | 0.5 | 1.0 | 1.0 |
| Robot Programming | LT | TP | O | OF | TO |

RE: Rectangular; CY: Cylinderical; SP: Spherical; AR: Articulated

NS: Non-servo; PTP: Servo Point-to-Point; CP: Servo Continious Path; PTP & CP: Combined PTP and CP

H: Hydraulic; EL: Electric; PN: Pneumatic

LT: Lead through teach Programming; TP: Teach-pendant Programming; O: On-line Programming; OF: Off-line Programming; TO: Task-oriented Programming

*B. Methodology for robot selection*

This system consists of several major modules as shown in the selection process architecture in Figure 4.3. The retrieval mechanism consists of:

- The structure of the case,
- Concept of similarity, and
- Semantic of taxonomy

This retrieval mechanism is used to measure the similarity between the present problem and previous cases already in the record. In general, the robot selection criteria include some key specifications such as; degrees of freedom, pay load, swept area, maximum reach, maximum speed, cost, and repeatability. Though most of the attributes have been identified, all of them may not be important for the intended application. There will be few attributes, which will have direct effect on the selection procedure. Some of the attributes may be selected as 'pertinent attributes' as necessitated by the particular application and/or the user. The threshold values to these 'pertinent attributes' are assigned by obtaining information from the user and the group of experts. The selection procedure focuses solely on the pertinent attributes leaving out the rest. On the basis of the threshold values, a shortlist of robots is obtained. This is achieved by scanning the database for those attributes, one at a time and eliminating the robot alternatives, which have one or more of these attribute values that fall short of threshold values. The robot selection architecture system is divided into five activities;

i) Initial operation survey,
ii) Operation qualification,
iii) Robot selection, and
iv) Robotic workcell Engineering and
v) Robotic workcell implementation.

The Continuous-type similarity measurement with the features like cost difference, price etc .are shown in Figure 4.4.

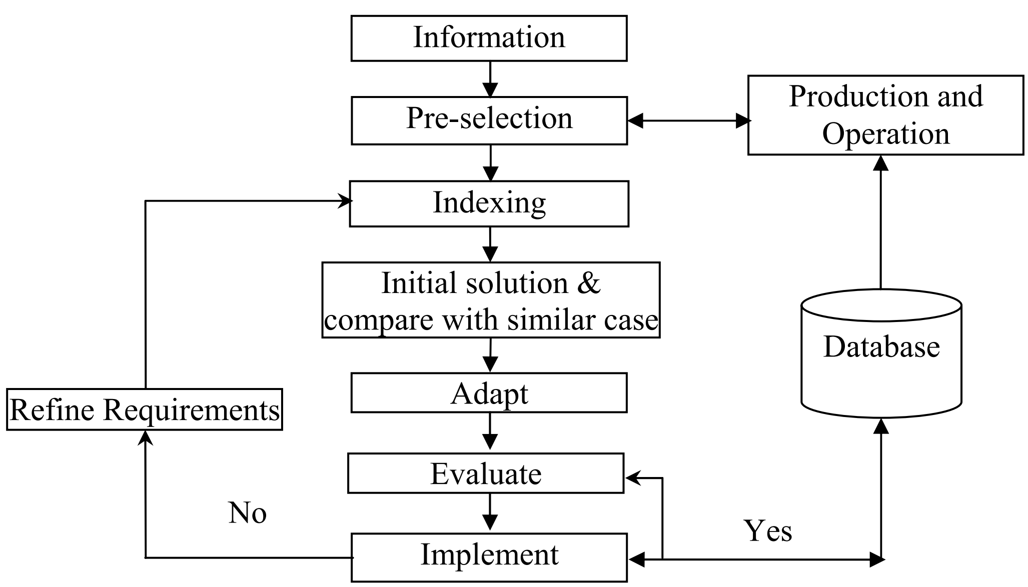



Figure 4.3 Robot selection process architecture

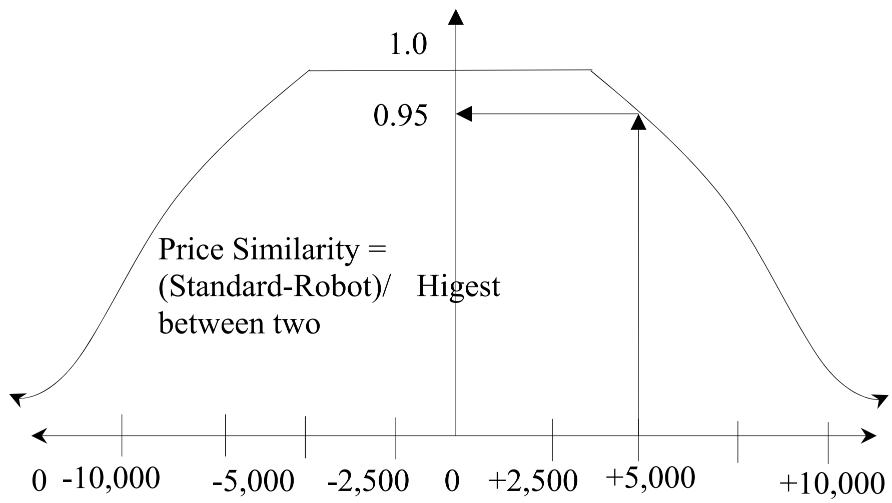



Figure 4.4 Similarity for the price difference of two robots

A few similarity measurement methods are considered in the present work. It is obvious that they have to be treated differently for the similarity computation. The current situation during the elaboration of a selection process is described by the information features known at the time point. The final problem, as it later may appear in the case memory, is a completed set of information features. The collected information features result form the real implementation case study.The geometry value of different types of robot are given in Table 4.12. Table 4.13 show a query problem and confirmed case for palletizing robot selection. Different features have different importance (weights). The following weights are considered for the similarity computation:

- Very important = 10 (Ex: repeatability, production rate, , etc);
- Important = 6 (Ex: payload, Degrees of freedom, etc);
- Somewhat important = 3 (Ex: velocity); and
- Unimportant = 1 (Ex: arm geometry)

For easy reference, all useful notations for our model are summarized in Table 4.15.

Table 4.12 Geometrical value of different robots

| Robot arm geometry | Cartesian | Cylinderical | Spherical | Articulated |
|---|---|---|---|---|
| Cartesian | 1.0 | 0.6 | 0.4 | 0.2 |
| Cylinderical | 0.6 | 1.0 | 0.6 | 0.4 |
| Spherical | 0.4 | 0.6 | 1.0 | 0.6 |
| Articulated | 0.2 | 0.4 | 0.6 | 1.0 |

Table 4.13 A simplified example 1

| Query problem | | | |
|---|---|---|---|
| Operation features | | Feature values | Weight |
| 1 | Production rate: | 480 task/hour | 10 |
| 2 | Operational Apple. Spec | Palletizing | 6 |
| | Repeatability | R ≤ ± 2 mm | 10 |
| | Work envelop (reach) | 2600 mm /102.3 in | 10 |
| | Payload | 100 kg / 220 lbs | 6 |
| | Velocity | S > 5000 mm | 3 |
| | Arm geometry | Articulated robot | 1 |
| | Degrees of freedom | 6 | 6 |
| | price | $ 100,000 | 10 |
| 3 | End-of-tooling Spec.: | Sucking cup type | 6 |
| 4 | Complexity of the task | 15 different boxes | 1 |
| 5 | Layout and space remit. | 16 $m^2$ | 3 |

Table 4.14 Similarity computation for example-1

| Features | Q | Similarity | Confirmed case 1 |
|---|---|---|---|
| price | \$100,000 | 0.952 | \$105,000 |
| Production rate: | 480 task/hour | 0.872 | 550 boxes/hour |
| Repeatability | R ≤ ± 2 mm | 0.5 | R ≤ ± 1 mm |
| Work envelop | 2600 mm | 0.923 | 2400mm |
| Payload | 100 kg | 0.83 | 120kg |
| Velocity | S > 5000 mm | 0.909 | 5500mm/s |
| Degrees of freedom | 5 | 0.833 | 6 |
| Arm geometry | Cartesian | 0.2 | Articulated |

Table 4.15 Notation for case

| Notations | Definition |
|---|---|
| *C'* | Confirmed case |
| *i* | Number of cases |
| *Q* | Query case |
| *I* | Factor index set, I = {1, 2, ..., n} |
| *n* | Number of factors to be shortlisted |
| S' | Similarity case |
| *J* | Factor index set,J={ 1,2,…,n} |
| *M* | weight |

Similarity for individual features $(S') = 1 - ((C_i - Q)/C'_i))$

Similarity query case= $1 / \sum M_I \left( \sum_{I=1, J=1}^{n} (M_I * S'_J) \right)$

Similarity (Case 1)= 1/56 [10 × 0.872 + 10×0.5 + 10 ×0.923 + 10×0.952 + 6 × 0.83 +3 ×0.909 + 6 × 0.833+1 ×0.2]= 45.375/56 = 0.810

Similarity (Case 2)= 1/56 [10 × 0.97 + 10×0.872 + 10 ×0.5 + 10 × 0.923 +6 ×0.833 +3 ×0.909 + 6 × 1.0 + 1×1.0 ]= 47.305/56 = 0.845

Similarity (Case 3) = 1/56 [10 × 0.952 + 10×0.872 + 10 ×0.5 + 10 × 0.923 +6 ×0.833 +3 ×1.0 + 6 × 1.0 + 0.4×1.0 ] =46.868/56=0.836

Similarity (Case 4) = 1/56 [10 × 0.952 + 10×0.872 + 10 ×0.5 + 10 × 0.846 +6 ×0.833 +3 ×0.925 + 6 × 0.838 + 0.2×1.0 ]=44.671/56=0.798

Similarity (Case 5) = 1/56 [10 × 0.97 + 10×0.872 + 10 ×0.5 + 10 × 0.846 +6 ×0.833 +3 ×0.909 + 6 × 1.0 + 1.0×1.0 ] =46.605/56=0.832

Table 4.16 Similarity computation example -2

| Features | Query Case | Similarity | Confirmed Case 2 |
|---|---|---|---|
| price | \$100,000 | 0.97 | \$103,000 |
| Production rate | 480 task/hour | 0.872 | 550 task /hour |
| Repeatability | R ≤ ± 2 mm | 0.5 | R ≤ ± 1 mm |
| Work envelop | 2600 mm | 0.923 | 2400mm |
| Payload | 100 kg | 0.833 | 120kg |
| Velocity | S > 5000 mm | 0.909 | 5500mm/s |
| Degrees of freedom | 5 | 1 | 5 |
| Arm Geometry | Cartesian | 1 | Cartesian |

For the purpose of simplification, we suggest to compute the global similarity between two cases based on the weighted sum of local similarity from all the robot features shown in Table 4.14 and Table 4.16.

## 4.5 Overview of task assignment

The most significant concept in MRS is cooperation. It is only through cooperative task performance that the superiority of robot groups can be demonstrated. The cooperation of robots in a group can be classified into two categories of implicit cooperation and explicit cooperation. In the implicit cooperation case each robots performs individual tasks, while the collection of these tasks is toward a unified mission. For example, when multiple robots are engaged in collecting rock samples and returning them to a common place, the team is accomplishing a global mission while cooperating implicitly. This type of group behavior is also called asynchronous cooperation, as it required no synchronization in time or space. The explicit cooperation is the case where robots in a team work synchronously with respect to time or space in order to achieve a goal. One example of such cooperation is transportation of heavy objects by multiple robots, each having to contribute to the lifting and moving of the object. This task requires the robots to be positioned suitably with respect to each other and to function simultaneously. Regardless of the type of cooperation, the goal of the team must be transformed into tasks to be allocated to the individual robots.

### 4.6 Summary

This chapter makes some contributions for robot selection. Firstly, it uses the fittness model based on attribute based theory for robot selection, which in turn allows the selected robot a better one as per the decision matrix. Secondly, the capability model takes into account the weight matrix for robot selection. The robots are selected as per the ranking results. Thirdly, the task requirement based model for selection of robots is developed for ranking of robots by combining manipulator attributes and task requirements in a comprehensive manner. The case based model takes into account a standard problem and compares with the robots' parameters as usually specified by the manufacturers which are used for ranking of the candidate robots.

# Chapter-V

# TASK ASSIGNMENT IN MRS

# CHAPTER 5

## Task Allocation in MRS

### 5.1 Introduction

As a result of the growing focus on MRS, multi-robot coordination has received significant attention. In particular, MRTA has recently risen to prominence and become a key research topic in its own right. Task means a subgoal that is necessary for achieving the overall goal of the system, and that can be achieved independently of other subgoals (i.e., tasks). Tasks can be discrete or continuous and can also vary in a number of other ways, including time scale, complexity, and specificity. Task independence is a strong assumption, and one that clearly limits the scope of this study. For example, ordering constraints on a set of tasks are not allowed, in general it is required that individual tasks can be considered and assigned independently of each other. The approach presented in the present work can be advantageously used in real-world problems.

An attempt is made to empirically derive some guidelines for selecting task allocation strategies for MRS. The allocation model is equivalent to a two-dimensional multi-type bin packing problem. Mathematical models and solution algorithms are presented for abetting task allocation in multirobot environment for accomplishing tasks.The explored strategies are individualistic in that they do not involve explicit coordination and negotiation among the robots. However, they are a part of a large class approaches that produce coherent and efficient cooperative behavior. The work aims at proposing a methodology to allocate tasks to available multiple type robots based on their capacity, availability and allocation cost. The focus here is on the development and implementation of an optimization algorithm for solving allocation model. Specifically, the objective of this work is to develop a solution algorithm that can be used to solve problems of a practical size within acceptable computational times. The characteristics of the allocation model warrant the development of an off-line algorithm. Although the procedures described here are in the context of robotics, these are general and applicable to any real-world application.

### 5.2 Allocation model formulation

One of the most important aspects in the design of MRS is the allocation of tasks among the robots in a productive and efficient manner. An empirical study is described for task allocation strategies. In general, optimal solutions are found through an exhaustive search, but because there are nxm ways in which m tasks can be assigned to n robots, an exhaustive search is often not possible with increased number of tasks. Task allocation methodologies must ensure that not only the global mission is achieved, but also the tasks are well distributed among the robots. The capability in terms of time and space are considered in the task allocation methodologies for MRS.

The task allocation approach considers the available resources, the capabilities of the deployable robots, and then it appropriately allocates the tasks to the candidate robots. Different approaches are presented and their results are analyzed for the suitability of the methods for an allocation problem.

Historical research in bin packing has focused on optimization problems involve some resource, and the task for algorithm designers is typically to get the job done using the minimum amount of resources. Bin packing is the problem of packing items of sizes between zero and one in the smallest possible number of bins of unit size [91, 92, 93]. Numerous investigators [94, 95] have examined the performance analysis of approximation algorithms designed for a number of two-dimensional bin packing variants. Optimal procedures for the constrained two-dimensional cutting problem have been proposed by several authors [96, 97]. Heuristic procedures for the constrained two-dimensional cutting problem have also been developed for using a problem representation which encodes the order in which pieces should be cut. The multi-choice multidimensional knapsack problem [98] is a combinatorial optimization problem. Given a set of groups of variables, one tries to select the best variable in each group. Both types of algorithms are practical in real-world applications.

The problem explicitly addresses robots of different types with various service time and space capacities. The assignment model seeks an optimal selection of robots to perform all given tasks such that each task's resource demands are satisfied, no robot's capacity constraints are violated, and the total system cost is minimized. A mathematical model along with its solution procedure is presented for allocation of tasks to the robots which is efficient and can serve as a planning tool. The model is formulated as a pure 0-1 mathematical program. Although, the key parameters for the model can be categorized as geometrical, kinematic, dynamic, power and noise, and thermal, the two most important factors while assigning tasks to robots are the geometrical work envelope and the kinematic machine cycle time. The work envelope for a typical robot is represented by a diameter of a circle. However, for the present model, it is not required that the work envelope be a complete circle. The time requirement of any task depends upon its relative distance from the robot.

The task can be represented by a workstation located at a definite distance from the robot and occupying a certain amount of space. In addition, the space requirement of a workstation also depends upon its relative location. If the workstation is assigned to location nearer to the robot its space requirement is smaller than what is required if assigned at location one. In contrast, the time requirement of a workstation assigned at location farther to robot is smaller than the time requirement associated with location which is nearer from robot because the latter incurs a longer travel time. Thus, there exists a trade-off between the space requirement and machine cycle time requirement. In fact, both the requirements are a function of the workstation's relative position from the robot. The primary objective is to minimize the total robot acquisition costs while satisfying workstation resource demands. In order to make the assignment model computationally tractable, it is assumed that all workstations are placed at the most remote location within the work envelope of the robot(s). This assumption decouples the interaction between space and time by allowing the resource requirements of a given workstation to be constant. Without this assumption, the model complexity is significantly increased. This trade-off between the number of robots required serving a given set of workstations and the time required to serve a workstation could be considered by iteratively solving the assignment model. The formulation of the model is as follows.

A set of robot types indexed by K = {1, 2, …, $k$}, is considered where each robot type is characterized by its time and space capacity. Specifically, space is measured in terms of the work envelope's swept area. The swept area is the total number of degrees around the central vertical axis that is within reach of the robot arm. All given workstations are indexed by I = {1, 2, …, $n$}.

Table 5.1 Notation for assignment model

| Notation | Definition |
|---|---|
| $k$ | Number of robot types |
| K | Robot type index set, K = {1, 2, ..., $k$} |
| $n$ | Number of workstations |
| I | Workstation index set, I = {1, 2, ..., $n$} |
| $m_k$ | Maximum number of type $k$ robots $\forall k \in K$ |
| $t_{ik}$ | Normalized time requirement of workstation $i$ when served by a type $k$ robot, $\forall\, i \in I, k \in K$ |
| $s_{ik}$ | Normalized space requirement of workstation $i$ when served by a type $k$ robot, $\forall\, i \in I, k \in K$ |

Each workstation $i$ demand a known amount of time and space when served by robot type $k$, denoted by $t_{ik}$ and $s_{ik}$ respectively. In addition, for a given set of $n$ workstations, let $m_k$ denote the maximum number of robots of type $k$ necessary to serve all workstations assuming only robots of type $k$ are available. Further, let $K$ = {1, 2, ..., $k$} denote the index set for type $k$ robots. For easy reference, all useful notations for the model are summarized in Table 5.1. A decision variable, $x_{ik}$, is defined as:

$$x_{ik} = \begin{cases} 1 & \textit{if workstatio n i is assigned to robot of type k} \\ 0 & \textit{otherwise} \end{cases}$$

With no loss of generality, the time and space requirements for each workstation (i.e., $t_{ik}$, and $s_{ik}$ , respectively) can be normalized by dividing the robot resource capacities into the corresponding workstation resource demands. This macro planning model does not consider variable costs and the solution algorithm developed is general. The robot selection and assignment (RSA) can be written as equation (5.1) through equation (5.5):

$$\text{(RSA) MIN} Z = \left( \sum_{k \in K} f_k \left( \sum_{k \in k} x_{ik} \right) \right) \tag{5.1}$$

Such that

$$\sum_{k \in K} \sum_{k \in k} x_{ik} = 1 \quad \forall i \in I \tag{5.2}$$

$$\sum_{i \in I} t_{ik} x_{ik} = 1 \quad \forall k \in K \tag{5.3}$$

$$\sum_{i \in I} s_{ik} x_{ik} \le 1 \quad \forall k \in K \tag{5.4}$$

$$x_{ik} \in \{0,1\} \quad \forall i \in I, k \in K \tag{5.5}$$

Condition (5.2) ensures that each workstation $i$ is assigned to exactly one robot. Conditions (5.3) and (5.4) ensure that workstations assigned to any robot will not violate the corresponding time and space constraints. The model is a pure 0-1 integer program (IP). Therefore, it is impractical to directly solve model by using any available IP code. Two different routes are followed to solve the model. In the first method, an optimization algorithm based on a greedy heuristic covering all the necessary parameters is developed for solving the task assignment problem in a heterogeneous multirobot environment.

## 5.3 Task analysis

The problem of task assignment in multirobot environment has been conceived with fifteen workstations and four robots. The robots under consideration are standard industrial robots. The pertinent parameters of the robots such as the number of DOF, pay load, swept area, maximum reach, maximum speed, type and cost are different. For the different tasks, normalized time and normalized space requirements are considered and then the load balance factor is determined. According to capabilities of the robots and requirement of the tasks some combinations of the robot and task may be time intensive whereas some other become space intensive. Then the adjusted demand is determined as per the assignment heuristic. The cost of allocation is determined by the product of cost of robots and adjusted demand. Before allocation of tasks to the robots, it is important to determine the loading capacity of the robots. The loading capacity of a robot depends on its individual capability which can be generally determined by its reach, speed, and pay load specifications. Besides the robot's capability, the loading capacity also depends on the requirement of the task (e.g. kind of operation, motion, and dimensions) and its location in the workspace. The methodology determines the loading capacity through the load deviation ratio which uses the normalized time and space requirement for various combinations of robot-task. The load deviation ratio encompasses all the required parameters for deciding the loading capacity. Load deviation ratio is the ratio of difference between the normalized space requirement ($s_{i1}$) and normalized time requirement ($t_{i1}$) to the summation of normalized space requirement ($s_{i1}$) and normalized time requirement ($t_{i1}$). After taking into account the absolute value of the load deviation ratio, the capacity of each individual robot is determined.

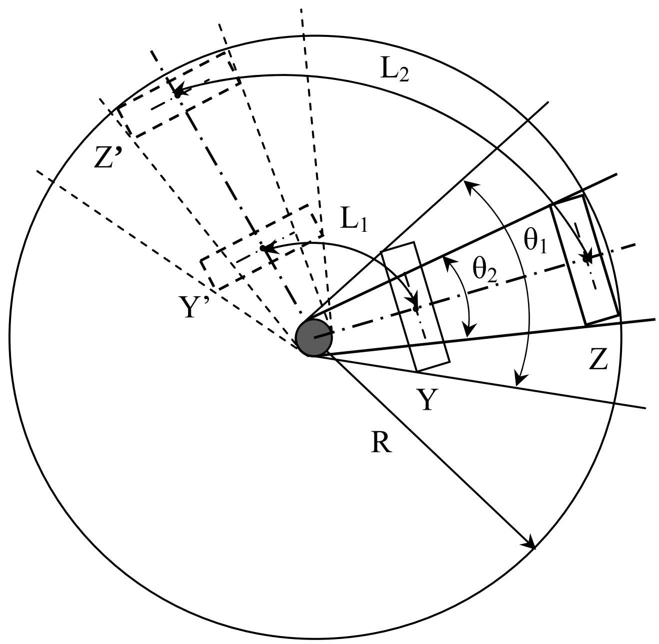
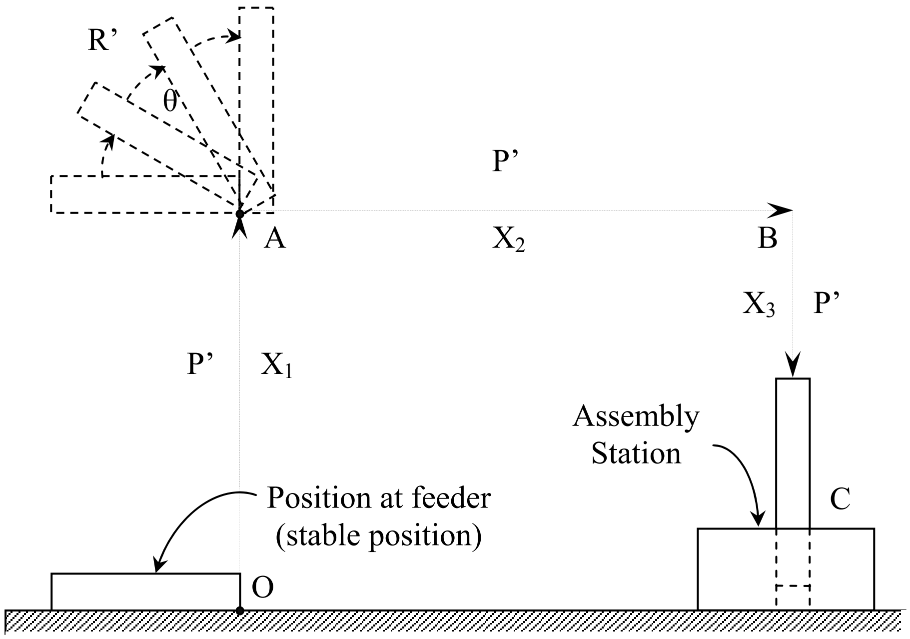



Figure 5.1 Time and space requirement for handling task

The time and space to handle a task are dependent upon the positions of the initial point and the target point of the part within the available workspace of a robot. To explain this point in a better way, a diagram is shown in Figure 5.1. Let the initial position of an object is at Y and the target point is at Z'. If the object is situated at Y', the work envelop is covering an area at an angle $\theta_1$ is $\pi R^2/\ \theta_1$ (R is the maximum reach). That means the robot handle the task within that area. The angle $\theta_i$ ($i = 1, 2$) is in 'rad'. If the object is moved to a distance along that centerline at position B, the covering area is reduced to $\pi R^2/\ \theta_2$, where $\theta_2 < \theta_1$. But the handling distance is increased from the center point. Again during the assembly if there is change of angular displacement $L_i$ ($i = 1, 2$), then the work coverage at arc ZZ' is more than that of YY'. This directly increases the time of assembly and space.



*P' is Prismatic motion and R' is revolute motion needed during the Task

Figure 5.2 Analysis of motion requirement of handling a task

An example of how an object is handled by a robot and what its time during assembly is shown in Figure 5.2. After picking up from the feeder the object is moved to assembly station. The assembly work is done at that station. In the example, there is four positions of movements and they are; O, A, B, C. 'O' is the feeder position. From O to A the robot needs prismatic motion. At A there is change in orientation at an angle θ; here it needs the rotary motion. And the last two positions it needs the prismatic motion.

Hence, the total time of task is the addition of time taken for distances $X_i$ ($i = 1, 2, 3$) and the rotational time. The time to complete one task is dependent upon the angular velocity. Once the layout of the various stations as dictated by the assembly sequence and the consequent allocation are decided, the distance of travel and the orientation is constant for a particular part. The addition of all the time to assemble a sequence of parts of a product is called the cycle time, which is considered as a candidate for minimization.

## 5.4 Assignment Heuristic (AH)

Given any workstation, three possibilities exist. The workstation can be time intensive, space intensive or neither. A heuristic is developed by examining these three cases, and the load balance on each candidate robot. The AH is based on the concept of allocation cost, which is computed as a function of the resource demands of each workstation and a robot's load balance. Let $\Delta_k$ denote the load balance factor associated with the robot of type *k*. That is, $\Delta_k$ is defined as the difference between the total allocated (normalized) machine time and the total allocated (normalized) work space for robot of type *k*. Let and where $x_{ik}$ is a 0-1 variable. Hence, $\Delta_k$ can be expressed as follows

$$\Delta_k = T_k - S_k \quad \forall k \in K \tag{5.6}$$

The following conditions are adopted for the model.

(i) If a robot’s resource load is nearly balanced, then the load balance factor will be approximately zero.

(ii) If the robot's load is time intensive, then $0 < \Delta_k < 1$, and

(iii) If the robot's load is space intensive, then $-1 < \Delta_k < 0$.

Hence, the further the resource load factor is away from zero, the greater the load imbalance is. In addition, let $\delta_{ik}$ denote the adjusted demand when the $i^{\text{th}}$ workstation is served by the robot of type *k*. That is,

$$\delta_{ik} = \begin{cases} \text{MAX}\{t_{ik}, s_{ik} - \Delta_k\} \text{ if } \Delta_k > 0 \\ \text{MAX}\{t_{ik} + \Delta_k, s_{ik}\} \text{ if } \Delta_k \le 0 \end{cases} \tag{5.7}$$

Since $0 < t_{ik} \le 1$, $0 < s_{ik} \le 1$, and $-1 < \Delta_k < 1$, we know $0 < \delta_{ik} \le 1$.

To illustrate how the adjusted demand is employed by AH, consider two robots of type *k*, say A and B. Assume that $\Delta_{Ak} = 0.4$ and $\Delta_{Bk} = -3$. Therefore, robot A is time intensive. In order to improve the load balance for robot A, one should prefer the assignment of a workstation which is space intensive (i.e., $t_{ik} < s_{ik}$) to those which are time intensive. By contrast, for robot B, the assignment of workstations which are time intensive should be given preference over workstations which are space intensive. An example is given below for illustration. Suppose the workstation to be assigned next is time intensive; that is, $t_{ik}$= 0.3 and $s_{ik}$= 0.2. Also, assume both robot A and B have enough remaining time and space capacities to serve this candidate workstation. The goal of our assignment heuristic is to balance the resource load on each robot. Since the candidate workstation is time intensive, it should be assigned to a robot which is space intensive. Plugging the given figures into equation (8) and have $\delta_{iAk} = 0.3$ and $\delta_{iBk} = 0.2$. These adjusted demands, i.e., $\delta_{ik}$ contribute to the "allocation costs". In general, if the fixed cost of all robot types is equal, the workstation should be assigned to the robot which produces the smallest adjusted demand. Since, not all robots have equal fixed cost, the allocation cost, $a_{ik}$ incurred by the $i^{\text{th}}$ workstation when served by the robot of type *k* is the product of its adjusted demand and the fixed cost of the robot. That is,

$$a_{ik} = f_k * \delta_{ik} \tag{5.8}$$

Since $0 < \delta_{ik} \leq 1$, we know that $0 < a_{ik} \leq f_k$. Thus, $a_{ik}$ reflects the adjusted proportion of the fixed cost that workstation *i* incurs when it is assigned to robot of type *k.* The heuristic is used to produce a good feasible solution. For each robot type *k,* the heuristic calculates the load deviation ratios and sorts them into a nondecreasing order. These load deviation ratios indicate the balance between the time and space requirements of each workstation when served by each robot type *k.* Then, the AH is employed to assign workstations to robots based on the sorted load deviation ratios. Since AH is simple and efficient, it is rerun once more based on a nonincreasing order of load deviation ratios. Our computational results indicate that AH provides a very good feasible and optimal solution.

It is assumed that each robot is capable of estimating its fitness for every task it can perform. This estimation includes factors, which are both task and robot dependent.

## 5.5 The example problem

Using realistic data, the following example is provided to highlight the solution process for an Allocation Model (AM) problem. While Table 5.2 summarizes major parameter values for four different robot types, Table 5.3 presents the normalized space and time requirements of fifteen workstations.

Robot-4 has a fixed charge of \$60,000, a swept area of $320^{\circ}$, a maximum reach of 1250 mm, and an average arm movement speed of 3.09 m/sec. Each entry in column two of Table 3 provides the diameter (*D'*) of a circle encompassing the workstation. It is assumed that each workstation is placed at the most remote location within the work envelope. Therefore, the *D'* associated with each workstation is in fact a chord to the work envelope. Knowing the value of *D'* and the maximum reach (*R*) of a robot, we can derive the arc length subtended by a workstation, which is $R\theta$ *where* $\theta = 2\sin^{-1}(D^{'}/2R)$. Here, $\theta$ represents the workstation's space requirement in degrees. Using $\theta$ and the swept area (*S*), a workstation's normalized space requirement can be determined. Considering workstation one and robot type one, we have *D'* = 1.0 meter, *R* = 1.25 meters, and *S* = $320^{\circ}$. Using this data, we have $\theta = 47.15^{\circ}$ and thus $S_{11} = (47.15/320) = 0.147$. In contrast, the time requirement of a workstation can only be determined after a thorough motion study of robot. In this macro planning model, the time requirement for each workstation is estimated based on two major components:

i) Robot arm travel time;
ii) Robot service time.

Both components are normalized by the total available machine time, which in practice is defined by the time available during peak machine hours. Using the above data and the aforementioned optimization algorithm, the allocation model is optimally solved. To proceed with the solution for allocation model, all the options of employing the available robot types are tried. The load balance factor $\Delta_{jk}$ and the allocation cost for each option are determined. There is a clear indication that the individual robots are better suited for the tasks only on the basis of their allocation cost than any of their combinations. This is a problem specific condition and it largely depends on number of factors such as time and space requirement. In other words, this is mainly due to low value of workstation size and relatively high value of the speed of the robots. The load balance factor, time requirement, space requirement and allocation cost are considered for the assignment of the robots to the workstations in question.

Table 5.2 Fixed costs and parameter values of the robots

| | Robot-1 | Robot-2 | Robot-3 | Robot-4 |
|---|---|---|---|---|
| Specification | (Puma 560-c) | (Adept one XL) | Fanuc Arcmate Sr.R.J | Staubli RX 130B |
| DOF | 6 | 4 | 6 | 6 |
| Pay Load | 4 kg | 12 kg | 10 kg | 12 kg |
| Swept Area | $320^{\circ}$ | $270^{\circ}$ | $300^{\circ}$ | $320^{\circ}$ |
| Max. Reach | 878 mm | 800 mm | 1529 mm | 1250 mm |
| Max Speed | 1.0 m/sec | 1.2 m/sec | 3.60 m/sec | 3.09m /sec |
| Type | Jointed | Scara | Jointed | Jointed |
| Cost | $35,000 | $19,500 | $56,400 | $60,000 |

Table 5.3 Normalized space and time requirements of workstations

| Workstation | | Normalized space requirement | | | | Normalized time requirement | | | |
|---|---|---|---|---|---|---|---|---|---|
| No.(i) | Size(D) | R-1 | R-2 | R-3 | R-4 | R-1 | R-2 | R-3 | R-4 |
| | | $S_{i1}$ | $S_{i2}$ | $S_{i3}$ | $S_{i4}$ | $t_{i1}$ | $t_{i2}$ | $t_{i3}$ | $t_{i4}$ |
| 1 | 1.0 | 0.216 | 0.286 | 0.127 | 0.147 | 0.214 | 0.216 | 0.203 | 0.2 |
| 2 | 0.7 | 0.146 | 0.192 | 0.088 | 0.101 | 0.143 | 0.145 | 0.141 | 0.142 |
| 3 | 1.1 | 0.242 | 0.321 | 0.14 | 0.163 | 0.237 | 0.243 | 0.225 | 0.228 |
| 4 | 1.05 | 0.229 | 0.303 | 0.133 | 0.155 | 0.224 | 0.229 | 0.213 | 0.216 |
| 5 | 0.9 | 0.192 | 0.253 | 0.114 | 0.131 | 0.188 | 0.191 | 0.181 | 0.184 |
| 6 | 1.01 | 0.219 | 0.289 | 0.128 | 0.148 | 0.215 | 0.219 | 0.205 | 0.208 |
| 7 | 0.65 | 0.135 | 0.177 | 0.081 | 0.094 | 0.133 | 0.134 | 0.13 | 0.131 |
| 8 | 0.7 | 0.146 | 0.192 | 0.088 | 0.101 | 0.143 | 0.145 | 0.14 | 0.142 |
| 9 | 0.75 | 0.158 | 0.207 | 0.094 | 0.109 | 0.154 | 0.156 | 0.15 | 0.152 |
| 10 | 0.85 | 0.18 | 0.237 | 0.107 | 0.124 | 0.177 | 0.179 | 0.171 | 0.173 |
| 11 | 1.1 | 0.242 | 0.321 | 0.14 | 0.163 | 0.237 | 0.243 | 0.224 | 0.227 |
| 12 | 1.5 | 0.366 | 0.515 | 0.195 | 0.23 | 0.359 | 0.39 | 0.313 | 0.322 |
| 13 | 1.4 | 0.33 | 0.452 | 0.181 | 0.212 | 0.324 | 0.342 | 0.29 | 0.297 |
| 14 | 1.2 | 0.269 | 0.359 | 0.154 | 0.179 | 0.264 | 0.272 | 0.246 | 0.25 |
| 15 | 1.18 | 0.263 | 0.351 | 0.151 | 0.176 | 0.258 | 0.266 | 0.241 | 0.245 |

To proceed with the solution for allocation model, all the options of employing individual and/or combination of available robot types are tried. Table 5.4 provides the load balance factors calculated for four robots. There is a clear indication that the individual robots are better suited for the tasks only on the basis of their allocation cost than any of their combinations. Figure 5.3 shows the capacity curves of the four individual robots that decide the distribution of load between robots.

Table 5.4 Load deviation ratio (LDR)

| Task | Robot-1 | Robot-2 | Robot-3 | Robot-4 |
|---|---|---|---|---|
| | LDR | LDR | LDR | LDR |
| 1 | 0.004 | 0.139 | 0.23 | 0.153 |
| 2 | 0.01 | 0.139 | 0.231 | 0.169 |
| 3 | 0.01 | 0.138 | 0.232 | 0.166 |
| 4 | 0.011 | 0.139 | 0.231 | 0.164 |
| 5 | 0.01 | 0.139 | 0.227 | 0.168 |
| 6 | 0.009 | 0.137 | 0.231 | 0.169 |
| 7 | 0.007 | 0.138 | 0.232 | 0.164 |
| 8 | 0.01 | 0.139 | 0.228 | 0.169 |
| 9 | 0.012 | 0.14 | 0.229 | 0.165 |
| 10 | 0.008 | 0.139 | 0.23 | 0.165 |
| 11 | 0.01 | 0.138 | 0.23 | 0.164 |
| 12 | 0.009 | 0.138 | 0.232 | 0.167 |
| 13 | 0.009 | 0.138 | 0.231 | 0.167 |
| 14 | 0.009 | 0.137 | 0.23 | 0.166 |
| 15 | 0.009 | 0.137 | 0.229 | 0.164 |

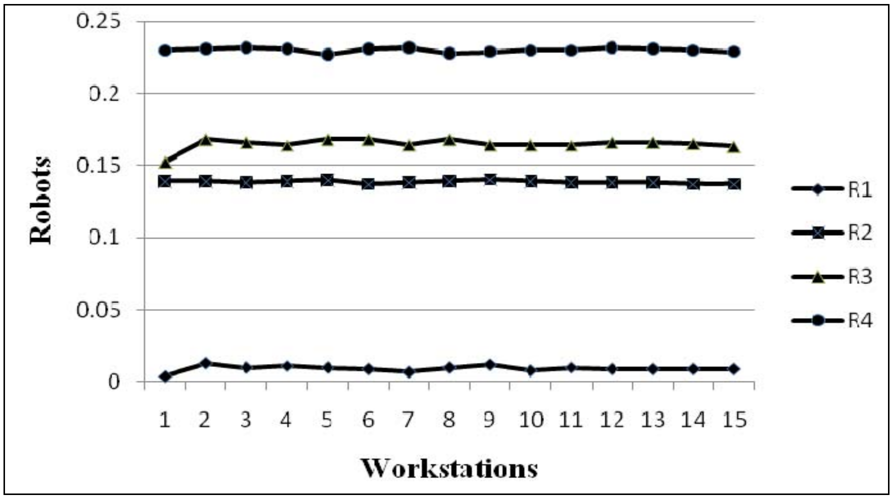



Figure 5.3 LDR of individual robot

Table 5.5 presents the allocation cost of the four robots for carrying out the designated tasks. The load balance factor, time requirement, space requirement and allocation cost are considered for the assignment of the robots to the tasks in question.

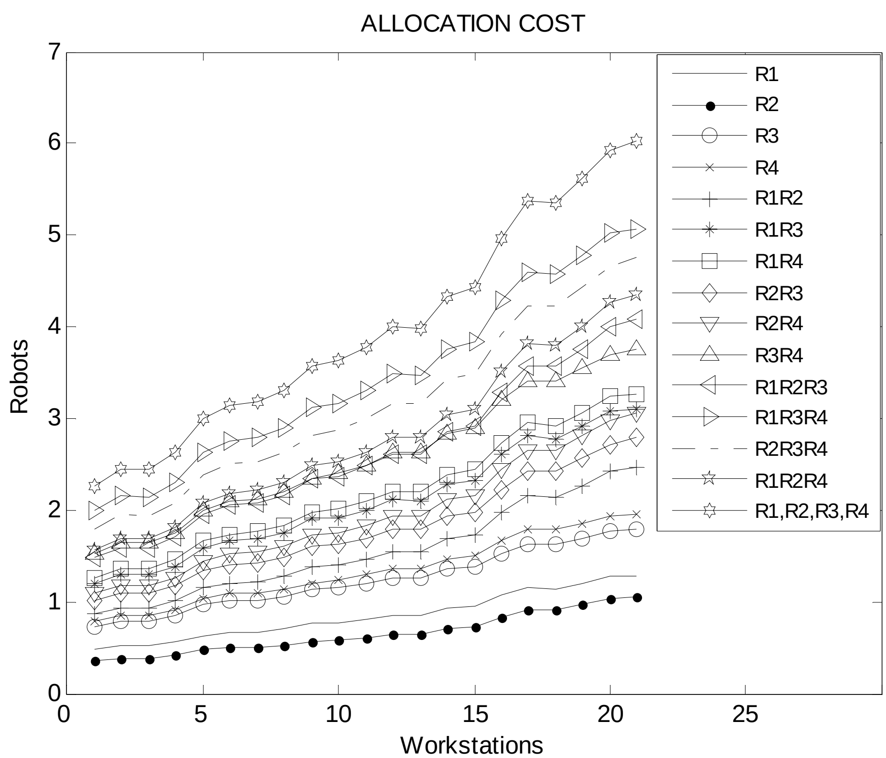



Figure 5.4 Allocation cost with all options

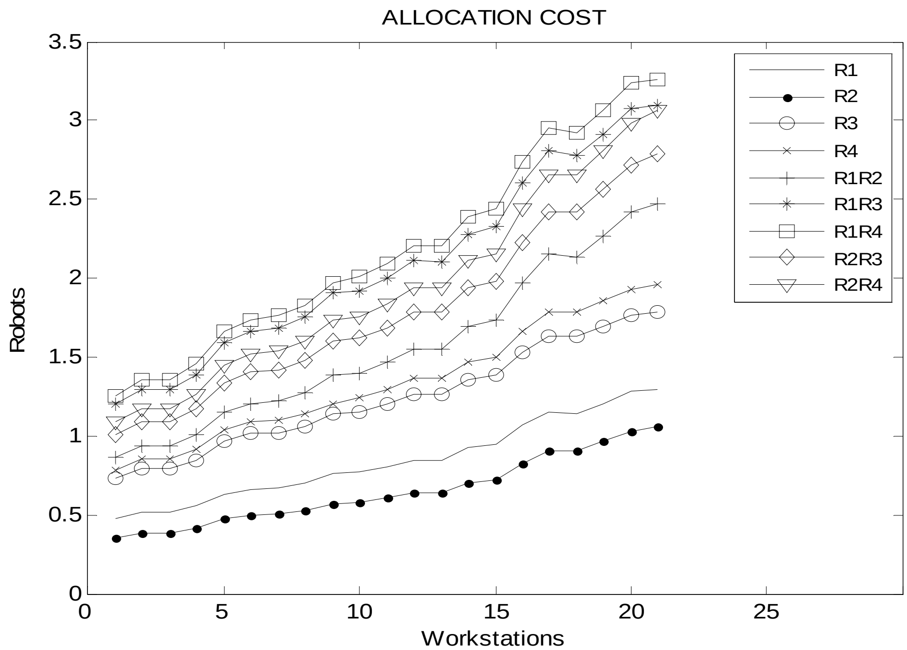



Figure 5.5 Allocation cost with truncated options

Figure 5.4 shows the allocation cost of all the 15 options of the robot combinations for the 21 workstations. However, on the relative allocation cost of six of the options (combination of robots for single task) came to be out of proportion and hence those options were left out of the set in the first instance. Figure 5.5 shows the allocation cost of the truncated 9 combinations. There is a clear indication that the individual robots are better suited for the tasks only on the basis of their allocation cost than any of their combinations.

Table 5.5 Allocation cost of assigned task

| Task | Robot-1 | Robot-2 | Robot-3 | Robot-4 |
|---|---|---|---|---|
| 1 | 0.475 | 0.355 | 0.733 | 0.786 |
| 2 | 0.513 | 0.384 | 0.795 | 0.852 |
| 3 | 0.513 | 0.384 | 0.79 | 0.852 |
| 4 | 0.553 | 0.414 | 0.846 | 0.912 |
| 5 | 0.592 | 0.444 | 0.902 | 0.972 |
| 6 | 0.633 | 0.475 | 0.964 | 1.038 |
| 7 | 0.658 | 0.5 | 1.015 | 1.086 |
| 8 | 0.674 | 0.507 | 1.021 | 1.104 |
| 9 | 0.699 | 0.526 | 1.06 | 1.14 |
| 10 | 0.759 | 0.573 | 1.145 | 1.2 |
| 11 | 0.767 | 0.579 | 1.156 | 1.248 |
| 12 | 0.803 | 0.607 | 1.201 | 1.296 |
| 13 | 0.848 | 0.642 | 1.269 | 1.368 |
| 14 | 0.848 | 0.643 | 1.263 | 1.362 |
| 15 | 0.923 | 0.703 | 1.359 | 1.47 |

## 5.6 Task assignment methods

With the advancement of technology, production systems are changing from traditional human dependent systems to intelligent automated systems. Industrial Robots have been instrumental in making the production systems more efficient, productive, responsive and flexible. In large production systems, multiple robots of different types, capacities and capabilities are employed for accomplishing the desired tasks. The flexibility and scalability of the system is greatly enhanced by use of multiple types of robots. The concept of using multiple robot types comes from the availability of those robots in the market. However the use of multiple type robots in a single workcell should not be done in random manner. It is desired that all the, devices in a workcell are controlled and coordinated properly through a single point (host) so that the workcell behaves like a single entity. Hence it is important to have the compatibility of the robots with the host. This calls for robot selection for the intended workcell. Since a multirobotic workcell is a cost intensive proposition the planning of such workcell should be done correctly. The selection of robots and subsequently the allocation of these robots for accomplishing the goal become prime issues in making the system efficient both from operation and economy view points. There is good number of tools available for optimizing the general allocation problems. However, if the robots under consideration are in large number possessing higher capability and the number of tasks to be carried out is large, then the number of alternatives for allocation becomes exorbitantly large, thereby making the allocation problem an NP-hard. Therefore, the optimization tool to be used for such problems need to be chosen carefully and correctly.

In order to treat task allocation in an optimization context, one must decide what exactly is to be optimized. Ideally the goal is to directly optimize overall system performance, but that quantity is often difficult to measure during system execution. Furthermore, when selecting among alternative task allocations, the impact on system performance of each option is usually not known. It is based on the notion that each individual can internally estimate the value (or the cost) of executing an action. Depending on the context, utility is also called fitness, valuation, and cost.

The different approaches are adopt and followed to optimize the assignment process. Several methods are available in texts for task assignment under various conditions. However, in view of the constraints and conditions existing in MRS, the following methods are picked up for task assignment to the robots. These are

i) Greedy Heuristics (GH)
ii) Linear Programming (LP)
iii) Mixed Integer Linear Programming (MILP)
iv) Knapsack Algorithm (KA)
v) Hungarian Algorithm (HA)
vi) Particle swarm optimization (PSO)

Once a particular task is assigned to a robot, the same is not considered for assignment to any other robots. The next step of the algorithm looks for the similar conditions as above amongst the rest of the tasks till the robots capacity constraints are satisfied.

### 5.6.1 Task assignment using Greedy Heuristics

In the first approach a greedy heuristic algorithm is used to assign tasks based upon the minimum allocation cost and the load deviation ratio. A GH is an algorithm that follows the problem solving metaheuristic of making the locally optimum choice at each stage with the hope of finding the global optimum. At each phase:

- You take the best you can get right now, without regard for future consequences
- You hope that by choosing a local optimum at each step, you will end up at a global optimum

In general, GH has five pillars:

i) A candidate set, from which a solution is created,

ii) A selection function, which chooses the best candidate to be added to the solution ,

iii) A feasibility function, that is used to determine if a candidate can be used to contribute to a solution,

iv) An objective function, which assigns a value to a solution, or a partial solution, and

v) A solution function, which will indicate when we have discovered a complete solution.

GH produces good solutions on some mathematical problems, but not on others. A GH may depend on choices made so far but not on future choices or all the solutions to the subproblem. It iteratively makes one greedy choice after another, reducing each given problem into a smaller one. In other words, a greedy algorithm never reconsiders its choices. This is the main difference from dynamic programming, which is exhaustive and is guaranteed to find the solution. After every stage, dynamic programming makes decisions based on all the decisions made in the previous stage, and may reconsider the previous stage's algorithmic path to solution. A problem exhibits optimal substructure if an optimal solution to the problem contains optimal solutions to the sub-problems. The solution algorithm for the assignment model uses a greedy heuristic-First Fit by Ordered Deviation (FFOD) to generate an initial feasible solution. The algorithm is used to search for the optimum. The heuristic provides an initial feasible solution which serves as an upper bound. This solution and its corresponding objective function value are then iteratively expanded and solved by using a decomposition procedure. This iterative solution process continues to refine the objective function [99,100]. If the final solution is all integers, then an optimal solution to the original assignment model problem has been found and the algorithm terminates. The tasks are carried out one after the other and for each task one robot is selected for carrying out the task. This algorithmic frame is presented in Table 5.6 and the details are given in A1 of Appendices.

Table 5.6 Algorithm frame for GH

| Algorithm |
|---|
| 1. J = Set of Jobs |
| 2. s = Ø |
| 3. while $J \neq Ø$ do |
| 4. choose $j \in J$ ; $J = J \setminus \{j\}$; |
| 5. choose $p \in P$; $s = s \cup (j,p)$ |
| 6. end while |
| 7. return s; |

In the heuristic, the task $j$ to be carried out next (line 4) and the robot $p$ to be selected for the task $j$ is scheduled (line 5). The following greedy approach for the selection of a robot for a task $j$ is followed.

### 5.6.2 Task assignment using Linear Programming

Cooperative control of multiple-robots is a complicated problem that requires real-time planning under communication constraints [101]. A cooperative controller must deal with a variety of problems such as sensor information, sub-team formation, and optimal task assignment; time/space coordinated control, and optimal trajectory generation. The present work focuses on the problem of task assignment, its formulation, and modeling. In fact, task assignment is a research topic studied for many years in the literature of operations research. However, task assignment in cooperative control requires online real-time solution. A single robot is able to service multiple tasks. Furthermore, some tasks must be serviced following a specific sequence in time. Therefore, task assignment for cooperative control is fundamentally different from off-line static task assignment studied in the literature. When a task is to be performed, it needs to be classified at the first instance and then its assignment is to be sought. Once a task is assigned, the task is viewed by other robots to ensure that it has been assigned. The tasks must be correctly assigned and distributed as per the load deviation ratio. Therefore, the task assignment in cooperative control is a dynamic process with changes, unexpected or expected, in the task in the system, and in the environment. The model of task assignment to multiple robots has been viewed as a problem for optimization using of LP technique. The software tool LINGO has been used to model and solve the problem. For creating a LINGO model, an optimization model consists of three parts:

- Objective function: This is single formula that describes exactly what the model should optimize: A general manufacturing example of an objective function would be to minimize the cycle time for a given product.
- Variables: These are the quantities that can be changed to produce the optimal value of the objective function.
- Constraints: These are formulas that define the limits on the values of the variables.

LP problems are mathematical programming formulations [102], where the objective and the constraints are linear functions of $\{X_1, X_2,\ldots,X_n\}$. Therefore, an LP formulation would look like this:

$$
\begin{aligned}
\text{Minimize} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \le b \\
& \ldots \quad \ldots \quad \ldots \\
& a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n \ge b \\
& \ldots \quad \ldots \quad \ldots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m
\end{aligned}
$$

where x represents the vector of variables (to be determined), while c and b are vectors of (known) coefficients and a is a (known) matrix of coefficients.

The problem of multiple task assignment is formulated using LP. The generalized LP formulation consists of n robots and m tasks so as to minimize the overall allocation cost. The cost matrix is for the problem has the size of 15×15, wherein 225 numbers of variables and 31 numbers of constraints are considered. All the variables with constraints are programmed in the LINGO software to optimize the objective function. The detailed programming is mentioned in A2 of Appendices.

### 5.6.3 Task Assignment using Mixed Integer Linear Programming

Mixed Integer Linear programs (MILP) techniques are effective not only for mixed problems like real and integer problems , but also for pure-integer problems, pure-binary problems, or in fact any combination of real, integer, and binary-valued variables. Fixed charges or set-up costs are incurred when there is some kind of fixed initial cost associated with the use of any amount of a variable, even a tiny amount. Fixed charges and set-up costs occur frequently in practice, so it is important to be able to model them.

While the MILP formulation is designed to be as flexible as possible to take many different parameters into account, the nature of MILP techniques makes it necessary to make a few simplifying assumptions in our problem model.

*Assumption 1: Job pre-emption is not allowed*

Once a robot begins to execute a task, it must continue to completion without interruption. Also, a robot may process only one task at a time.

*Assumption 2: All relevant parameters to the problem are known in advance*

To compute the task allocation to optimality, information about all available robots including services offered, execution delays, and communication delays are needed. If information about robots is not known *apriori*, those resources will not be taken into account when solving the problem.

*Assumption 3: The robot network is static*

Because the entire allocation is computed prior to execution, any dynamic changes in the robot organization or job precedence graph during execution of the schedule may result in a sub-optimal or infeasible solution.

*Assumption 4: Each individual task can be completed with a single robot*

If a task requires a combination of services from multiple robots to be completed, that job must be further decomposed into smaller tasks specific to each service before the optimal allocation is computed, or that group of robots must be modeled as a single robot. While these assumptions place some limits on the types of problems that can be solved using this technique, the algorithm is still flexible enough to be used for modeling task allocation problems in many different scenarios.

An MILP [103] is to coordinate multiple heterogeneous robots for detecting and controlling multiple regions of interest in an unknown environment. The objective function should contain four basic requirements of a multi-robot system serving this purpose: control regions of interest, provide communication between robots, control maximum area and detect regions of interest. This solution defines optimum locations of robots in order to maximize the objective function while efficiently satisfying some constraints such as avoiding obstacles and staying within the speed capabilities of the robots.

An approach is developed for solving the MRTA problem considering a reduced domain. A generalized problem is formulated and considering the robots under question in terms of their space and time capabilities and the requirement of tasks an initial solution is obtained on the number of tasks that can be allocated to the candidate robots. Thereafter, MILP technique is used to obtain the optimized MRTA.

An MILP approach is presented in the context of a MRTA problem framework that enables optimal makespans to be computed for complex classifications of scheduling problems taking multiple parameters into account. Many LP problems exist where it is necessary to restrict the decision variables to integer or binary values. Examples include cases where the decision variable represents a nonfractional entity such as people or bicycles, or where a decision variable is needed to model a logical statement (such as whether or not to assign task A to agent B). These problems are called MILP problems, and are often much harder to solve than LP problems. This is because instead of having feasible solution points at the easily computed corners of the feasible region, they are instead usually internal and more difficult to locate. For example, constraining X and Y from the LP formulation to have integer values, the feasible solution points are shown in Figure 5.6.The first step in solving a MILP problem such as this is to solve the linear relaxation of that problem. This simply means removing the constraints that any decision variables have integer values and solving the resulting LP problem using an algorithm such as the Simplex Method. The result is one of the following outcomes:

- The LP problem is infeasible, so the MILP problem is also infeasible.
- The LP is unbounded and is probably not a well-posed problem.
- The LP has a feasible solution and all integrality constraints are satisfied, so the MILP has also been solved.
- The LP has a feasible solution, but not all of the decision variables have integer values.

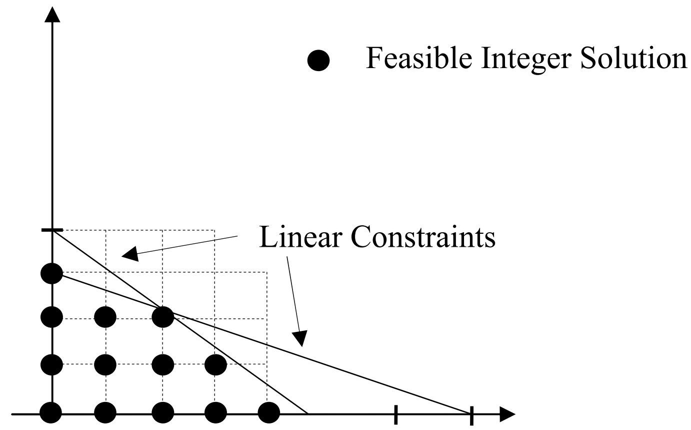



Figure 5.6 A MILP problems showing all feasible integer solutions

A new, flexible MILP is formulated that can be used to solve task allocation problems with a variety of parameters in the context of a multirobot problem solving framework. MILP techniques are chosen to model task allocation problems because of the intuitive nature of modeling these problems as a set of constraints with an objective function, and because these techniques produce optimal solutions. A significant amount of work has been done in developing and optimizing this MILP formulation by reducing the number of binary variables and redundant constraints used in the model. The algorithm uses MILP techniques to solve the most complicated classifications of the task allocation problem.

MILP technique has several advantages. Firstly, MILP produces exact optimal solutions instead of approximate ones. Secondly, being a general-purpose optimization method, software tools such as Management Scientist are available to efficiently solve an MILP problem once it has been formulated. The MILP approach offers more flexibility than most existing task allocation algorithms and heuristics.

The algorithm achieves completely optimal task allocation for multi-robot problems. The disadvantage of using MILP techniques is that they are NP-hard, and therefore may be infeasible to use for solving larger MRTA problem. Another drawback in using MILP algorithm is that since it runs in exponential time it is only feasible to use on smaller-scale problems, where there are a limited number of robots and tasks. As the algorithm is static rather than dynamic, all information about the robot network must be known *apriori* to solving the problem.

### 5.6.4 Task assignment using Knapsack Algorithm

The KA problem [104] considers each robot as a two-dimensional bin [105] and each task as a two-dimensional object to be packed. The model is viewed as assigning objects into an optimal set of bins such that both resource demands of each object are satisfied and neither of the capacity constraints of each selected bin is violated. This KA problem's solving goal is to find a subset of objects that maximizes the total profit while satisfying some resource constraints and allocation of task suitably. The problem of task allocation in MRS is modeled accordingly where, m is the number of robots and n is the number of tasks. Let p' be an n × m profit matrix. The value of p' [i, j] indicates the profit of task i when selected for robot j with capacity c'. The solution is a valid assignment since no item can be assigned to more than their capacity of robots. Given a set of tasks, each with a cost and a value, the objective is to determine the number of each task so that the total profit is maximized and the total value is as large as possible. The developed algorithm chooses the objective function and makes the allocation of task to individual robot. The 0-1 KA problem is posed as follows. For n tasks; the ith task is worth pj and weighs wj, where pj and wj are integers. The 0-1 KA wants to take as valuable a load as possible, but carry at most w in the knapsack for some integer w. This is called the 0-1 KA problem because each item must either be taken or left behind. The problem has 15 tasks, and the knapsack can hold profit of 15. The detailed procedure to solve the knapsack problems is mentioned in A3 of Appendices.

Considering a bounded amount $m_j$ of each item type j, the bounded KA problem arises as:

$$\text{Maximize} \quad \sum_{j=1}^{n} p'_j \, x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \le c',$$

$$x_j \in \{0,1,\ldots,m_j\},\ j = 1,\ldots,n.$$

$$S_{avk} = \frac{\sum_{i=1}^{n} S_{ik}}{\sum n};\ k = 1,\ldots,4$$

where, $S_{ik}$ is the space requirement of task when served by robot k, and *n*- number of tasks.

Considering unit weight for all the tasks, task 1 is worth 8 for robot-1. Similarly, worth of task 1 is 14, 5 and 5 for robot-2, robot-3 and robot-4 respectively. In the same way, the profit and capacity of robots for all fifteen tasks are determined and are presented in Table 5.7 and Table 5.8 respectively. The average task serving capacity may be taken as inverse of $S_{avk}$ i.e 1/ $S_{avk}$. Cost of individual robot, expected period of robot, average working hour per day and the time requirement of the task are taken into account to determine the profit function, The profit is assumed as the inverse of cost, where the cost is defined as;

$$Cost = \left( \frac{f_k}{\text{expected period} \times \text{working hour}} \right) \times t_{ik}$$

where $f_k$ is the cost function of a robot type

Table 5.7 The profit values

| Task | Robot-1 | Robot- 2 | Robot- 3 | Robot- 4 |
|---|---|---|---|---|
| 1 | 8 | 14 | 5 | 5 |
| 2 | 7 | 13 | 5 | 4 |
| 3 | 7 | 13 | 5 | 4 |
| 4 | 7 | 12 | 4 | 4 |
| 5 | 6 | 11 | 4 | 4 |
| 6 | 6 | 10 | 4 | 4 |
| 7 | 5 | 10 | 3 | 3 |
| 8 | 5 | 9 | 3 | 3 |
| 9 | 5 | 9 | 3 | 3 |
| 10 | 5 | 8 | 3 | 3 |
| 11 | 4 | 8 | 3 | 3 |
| 12 | 4 | 8 | 3 | 3 |
| 13 | 4 | 7 | 3 | 2 |
| 14 | 4 | 7 | 3 | 2 |
| 15 | 4 | 6 | 3 | 2 |

Table 5.8 Capacity of robots

| Robot-1 | Robot-2 | Robot-3 | Robot-4 |
|---|---|---|---|
| 3 | 3 | 5 | 4 |

### 5.6.5 Task assignment using Hungarian Algorithm

Before allocation of tasks to the robots, it is important to determine the loading capacity of the robots. The loading capacity of a robot depends on its individual capability which can be generally determined by its reach, speed, and pay load specifications. Besides the robot's capability, the loading capacity also depends on the requirement of the task (e.g. kind of operation, motion, and dimensions) and its location in the workspace. The loading capacity is determined through the load deviation ratio which uses the normalized time and space requirement for various combinations of robot-task. Load deviation ratio is the ratio of difference between the normalized space requirement ($s_{i1}$) and normalized time requirement ($t_{i1}$) to the summation of normalized space requirement ($s_{i1}$) and normalized time requirement ($t_{i1}$). After taking into account the absolute value of the load deviation ratio, the capacity of each individual robot is determined. As per LDR, the distributions of robot are balanced in HA. The popular and several restricted forms of task allocation issues are NP problems. It searches a feasible matching scheme to realize corresponding object models. The Hungarian algorithm [106,107] is a combinatorial optimization algorithm which solves assignment problems in polynomial time. The algorithm models an assignment problem as $n \times m$ cost matrix, where each element represents the cost of assigning the robot to the task. By default, the algorithm performs minimization on the elements of the matrix; hence in the case of a price-minimization problem, it is sufficient to begin Gaussian elimination to make zeros appear (at least one zero per line and per column). However, in the case of a profit-maximization problem, the cost matrix needs to be modified so that minimization of its elements results in maximizing the original cost values. In an infinite-cost problem, the initial cost matrix can be re-modeled by subtracting every element of each line from the maximum value of the element of that line (or column respectively). In a finite-cost problem, all the elements are subtracted from the maximum value of the whole matrix. It resolves the problem for the robot how to get the tasks and realize them at minimal cost. In this algorithm the input is a cost table established according to the cost needed for completing different tasks, and the output is an equivalent cost table in which a complete assignment constitutes an optimal assignment. The main idea of the algorithm is to modify the cost table's columns and rows until there is at least one zero in every column or row so as to find an complete assignment scheme according to the zeroes. This scheme is an optimal assignment when it is applied to the cost matrix for the total cost in this scheme is the least, and the algorithm can be always converging on an optimal solution in finite steps. The basic theory of this algorithm is that when you add a constant to any row (column) or subtract a constant form any row (column), the optimal assignment will not change.

Hungarian algorithm is feasible in the multi robot system domain, and it can efficiently and evenly distribute tasks among the candidate robots. The solution steps are as follows:

Step l: Modify the cost matrix. First, subtract the smallest element in every row from the row, and then subtract the smallest element in every column from the column in the matrix so that there is at least one zero in every row and column.

Step 2: If there is a complete assignment scheme, a cost matrix is obtained with N zeroes in different columns and rows which is the optimal solution and the algorithm is over, else go to next step.

Step 3: Cover all the zeroes in the cost matrix with the least lines, then find the smallest element in the remaining matrix and subtract it from every element not covered and add it to the line-cross elements.

Step 4: If the zero elements in the matrix constitute a complete assignment, go to step 5, else go to step 3.

Step 5: Add the cost to the zero elements located, then the sum is the total cost and the assignment is the optimal one.

The Hungarian algorithm is used to realize task allocation of the robots based on two-dimensional assignment problem aiming at multi-robot system. Hungarian algorithm is used to realize the task allocation of the multi robot system and then compare it with other task allocation methods. The procedure of Hungarian Algorithm is mentioned in A4 of Appendices.

### 5.6.6 Task assignment using particle swarm optimization (PSO)

PSO is a population based stochastic optimization technique, inspired by social behavior of bird flocking or fish schooling, and is developed by Dr. Eberhart and Dr. Kennedy in 1995[108]. The system is initialized with a population of random solutions and searches for optima by updating generations. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. The advantages of PSO are easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.

As stated before, PSO simulates the behaviors of bird flocking. This can be explained by the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. The birds do not know where the food is. But they know how far the food is in each iteration. So the best strategy to find the food is to follow the bird which is nearest to the food. PSO is learnt from the scenario and this technique is used to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest. After finding the two best values, the particle updates its velocity and positions with following equation (5.1) and (5.2).

Velocity update : $v_i(t+1)=w\ v_i(t) + c1*rand*(pbest(t) - x_i(t)) + c2*rand * (gbest(t) - x_i(t))$ (5.1)

Position update: $x_i(t+1)=x_i(t) + v_i(t+1)$ (5.2)

Where

$w > (1 / 2) (C1 + C2) – 1$

$0 < w < 1$

Most of evolutionary techniques have the following procedure:

1. Random generation of an initial population
2. Reckoning of a fitness value for each subject. It will directly depend on the distance to the optimum.
3. Reproduction of the population based on fitness values.
4. If requirements are met, then stop. Otherwise go back to 2.

The PSO-based algorithm is the optimization problem for task allocation on the basis of global optimization. The problem that $n$ tasks on $m$ robots with an objective of minimizing the completion time and utilizing the resources effectively. If the number of tasks is less than the number of robots in dynamic environment, the tasks can be allocated on the robots according to the first-come-first-serve rule. If the number of task is more than the number of robots, the allocation of tasks is to be made as per the algorithm. Considering the number of tasks is more than the robots, one task cannot be assigned to different robots, implying that the task is not allowed to be migrated between robots. The aim of this problem is to improve the efficiency of robots and to minimize the completion time at the same time. PSO can be implemented to solve various function optimization problems. The code of PSO is mentioned in A5 of Appendices.

### 5.7 Integration of task allocation with task planning in MRS

In practice, the processes of robot selection and task assignment cannot be treated in isolation. These processes are to be considered in an integrated manner in order to perform the desired task. Hence, there should be an integrated approach towards this. An example problem of assembly under MRS is considered here to validate the suitability of methodologies already mentioned in the previous sections.

An assembly task involves joining two or more components or subassemblies together. An assembly plan for a given product consists of a set of assembly tasks with ordering constraints among its elements. Each assembly task consists of joining a set of subassemblies to yield a larger subassembly. Given an assembly plan, an assembly sequence is an ordered sequence of the assembly tasks that satisfies all the ordering constraints. Each assembly plan corresponds to one or more assembly sequences. Efficient manufacturing in industries is conditioned by assembly.

The problem explicitly addresses robots of different types with various DOF and capacities. The assignment model seeks an optimal selection of robots to perform all given tasks such that each task's resource demands are satisfied, no robot capacity constraints are violated, and the distribution is balanced. Multi-robot teamwork is a complex problem consisting of task division, task allocation, coordination, and communication. The most significant concept in multi-robot systems is cooperation. The problem of task assignment in multirobot environment has been conceived with twenty one parts with twenty two task assemblies and two robots.

The objective of the assembly plan is to minimize the total assembly time. The algorithm takes into account the consideration of robots specifications, the dexterity of robots and the motion requirements for accomplishing the tasks. To meet the objective the process starts from the disassembly completed graph. The best sequence is generated through the evolutionary computation (ant colony) technique by considering the assembly constraints, the in-process stability and the motion studies of the parts to be assembled. The evolution technique Ant Colony Optimization (ACO) is a model-based metaheuristic approach for solving hard combinatorial optimization problems. The term metaheuristic is a set of algorithms concepts that can be used to define heuristic methods applicable to a wide set of different problems, and combinatorial optimization problem is either a maximization or a minimization problem which has associated with a set of problem instances. The term instances refer to a problem with specified values for all the parameters. The inspiring source of ACO is the foraging behaviors of the real ants which enables them to find shortest path between a food source and their nest. Some other type of such evolutionary approaches are; SA, Neural Network, Evolutionary Computation, PSO, Artificial Immune System, and so on. The ACO algorithms have been applied successfully in a variety of optimization problems that can be expressed as searching for optimal paths on graphs, such as the traveling salesman problem (TSP), Just-in-time (JIT) sequencing, job-shop scheduling etc. It is best suitable for combinatorial optimization problems. During the construction of sequences in ACO, local pheromone updating encourages exploration of alternative solutions, while global pheromone updating encourages exploitation of the most promising solutions. The work in ACO is further extended to multirobot allocation model. The principle of sequential and parallel execution of tasks through the parameters involved in total assembly time is worked out. Once the sequence is generated it is time to implement the part sequences to a multi-robotic environment in the industries. Before evaluating the best possible combinations of robots assignment one criterion is taken into effect i.e. to maximize the amount of parallelism that is possible in the execution of the assembly tasks. This drastically reduces the total assembly time of a sequence. The methodology is developed on the concept of increase in system flexibility by shared manufacturing, material handling resources, and reduction in cycle time by concurrent work. Task-sharing and resource-sharing are repetitively investigated during the running of the problem. While allocating assembly tasks, the following three different options are taken into account;

Option 1: Task allocation to the robots are made on the basis of the robot's capability to fulfill the motion conditions is made on the basis of robot's capability to fulfill the motion conditions (type of its and DOFs);
Option 2: Tasks are allocated to robots alternatively as per the generated sequence;
Option 3: Tasks are allocated to robots in accordance with their capabilities and time availability.

### 5.7.1 Illustrative example

The robots under consideration are standard industrial robots. The two robots (PUMA 560 and Adept One Xl) have been selected on the basis of motion and stability requirements of this kind of problem. The aim of the selection is to reduce the total assembly time and optimize the robot selection sequence for assigning the tasks. The pertinent parameters of the robots such as the number of DOF, pay load, swept area, maximum reach, maximum speed, type and cost are different. The tasks are characterized with the movement and orientation values of the respective task sequences. According to capabilities of the robots and requirement of the tasks some task may be 6 DOF whereas some other becomes 4 DOF. Using realistic data, the following example is provided to highlight the solution process for an allocation problem. Puma 560 robot has a swept area of $320^{\circ}$, a maximum reach of 878 mm, and an average arm movement speed of 1.0 m/sec with rotational jointed type. Adept one XL robot has a swept area of $270^{\circ}$, a maximum reach of 800 mm, and an average arm movement speed of 1.2 m /sec with prismatic jointed type. For the different tasks, DOF requirements are considered and then the distribution is determined. Let us example task 7 is a screw is to assemble in task 16. To assemble the task 7 in task 16, there are requirements of 5 DOF. Finally PUMA 560 robot is assigned to task 7 because Adept one XL is only for 4 DOF. The detail procedure of finding the optimal allocation is represented as flow diagram in the Figure 5.7.

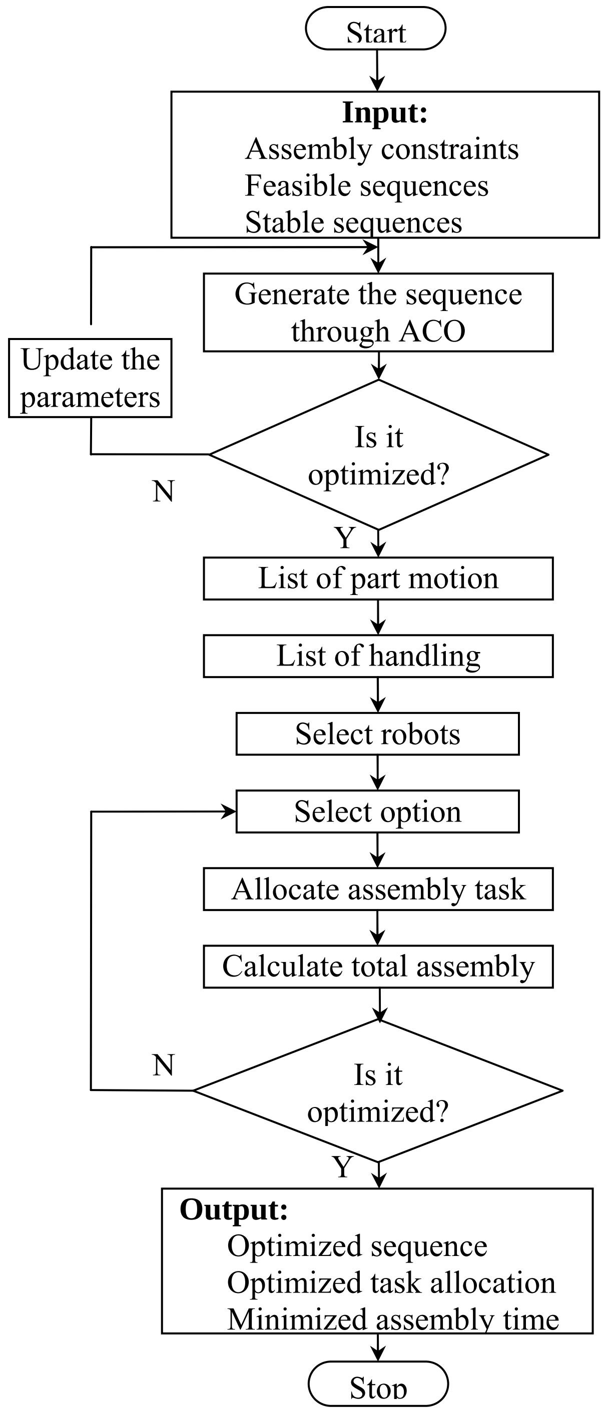



Figure 5.7 Flow chart of the proposed methodology

### 5.7.2 The assembly problem

In order to work on the proposed method, an example product (Drive Assembly) is considered. The assembly is an electro-motor device with casing used as a drive motor. The exploded view of the assembly is shown in Figure 5.8 having 21 numbers of parts including 10 screws. The part description and the part connections with are given in Table 5.9 and Table 5.10 respectively.

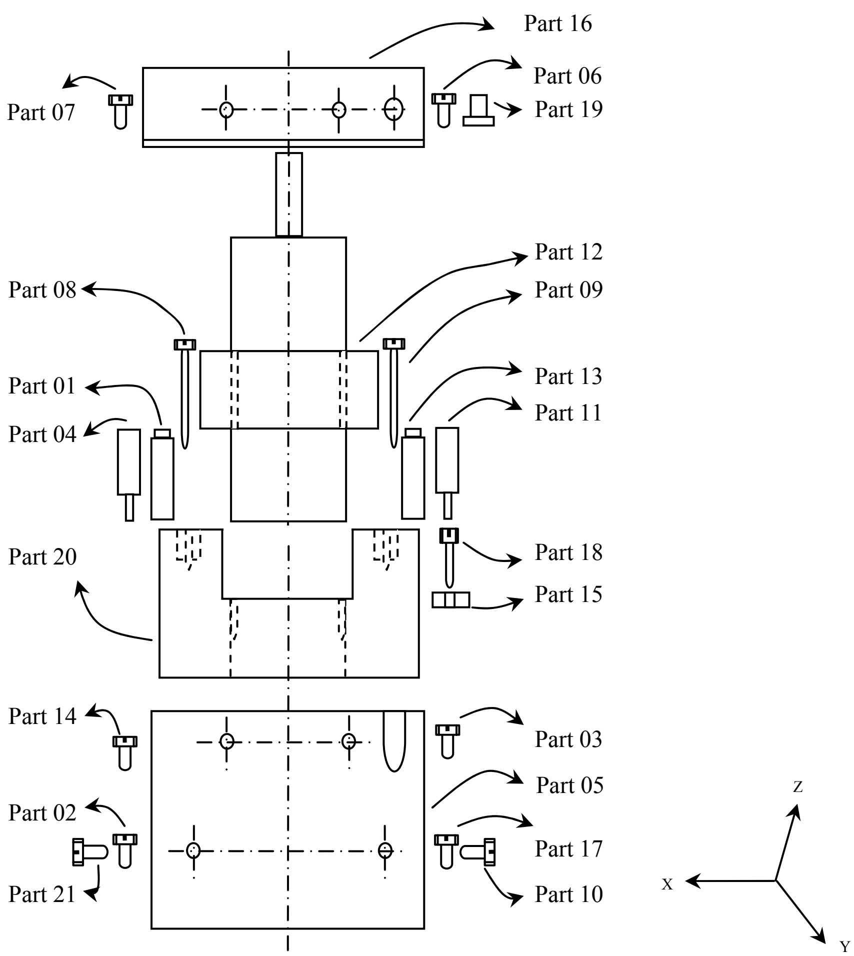



Figure 5.8 Exploded view of a drive assembly

Table 5.9 Part description

| Part Number | Part description |
|---|---|
| 1, 13 | Stud- I |
| 4, 11 | Stud-II |
| 2, 3, 6, 7, 10, 14, 17, 21 | Screw-I |
| 8, 9 | Screw-II |
| 18 | Screw-III |
| 5 | Shell |
| 12 | Electromotor |
| 15 | Washer |
| 16 | Cover Plate |
| 19 | Plug |
| 20 | Base |

Table 5.10 Part description with connectivity relation

| Part number | Connectivity relation with |
|---|---|
| 1 | 16, 20 |
| 2 | 05, 20 |
| 3 | 05, 16 |
| 4 | 07, 16, 20 |
| 5 | 02, 03, 10, 14, 16, 17, 19, 20, 21 |
| 6 | 11, 16 |
| 7 | 04, 16 |
| 8 | 12, 20 |
| 9 | 12, 20 |
| 10 | 05, 20 |
| 11 | 06, 16, 20 |
| 12 | 08, 09, 15, 16, 20 |
| 13 | 16, 20 |
| 14 | 05, 16 |
| 15 | 12, 18, 20 |
| 16 | 01, 03, 04, 05, 06, 07, 11, 12, 13, 14, 19 |
| 17 | 05, 20 |
| 18 | 15, 20 |
| 19 | 05, 16 |
| 20 | 01, 02, 04, 05, 08, 09, 10, 11, 12, 13, 15, 17, 18, 21 |
| 21 | 05, 20 |

As found from the liaison diagram shown in Figure 5.9 the product contains 38 numbers of liaisons. The dotted boundary in the diagram (Figure 5.9) envelopes a possible subassembly of the product. The liaisons can be represented as;

$$L_{01} = l_{04-07} = \left( 04, \begin{pmatrix} 0 & 0 & vc \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & sw \\ 0 & 0 & 0 \end{pmatrix}, 07 \right)$$

$$L_{02} = l_{07-16} = \left( 07, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & rc \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & rf \end{pmatrix}, 16 \right)$$

$$L_{03} = l_{04-16} = \left( 04, \begin{pmatrix} 0 & 0 & rc \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, 16 \right)$$

$$\cdots\cdots\cdots$$

$$\cdots\cdots\cdots$$

$$L_{36} = l_{15-18} = \left( 15, \begin{pmatrix} 0 & 0 & rc \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & sw \\ 0 & 0 & 0 \end{pmatrix}, 18 \right)$$

$$L_{37} = l_{18-20} = \left( 18, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & vc \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & sw \end{pmatrix}, 20 \right)$$

$$L_{38} = l_{05-14} = \left( 05, \begin{pmatrix} 0 & rc & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & sw & 0 \\ 0 & 0 & 0 \end{pmatrix}, 14 \right)$$

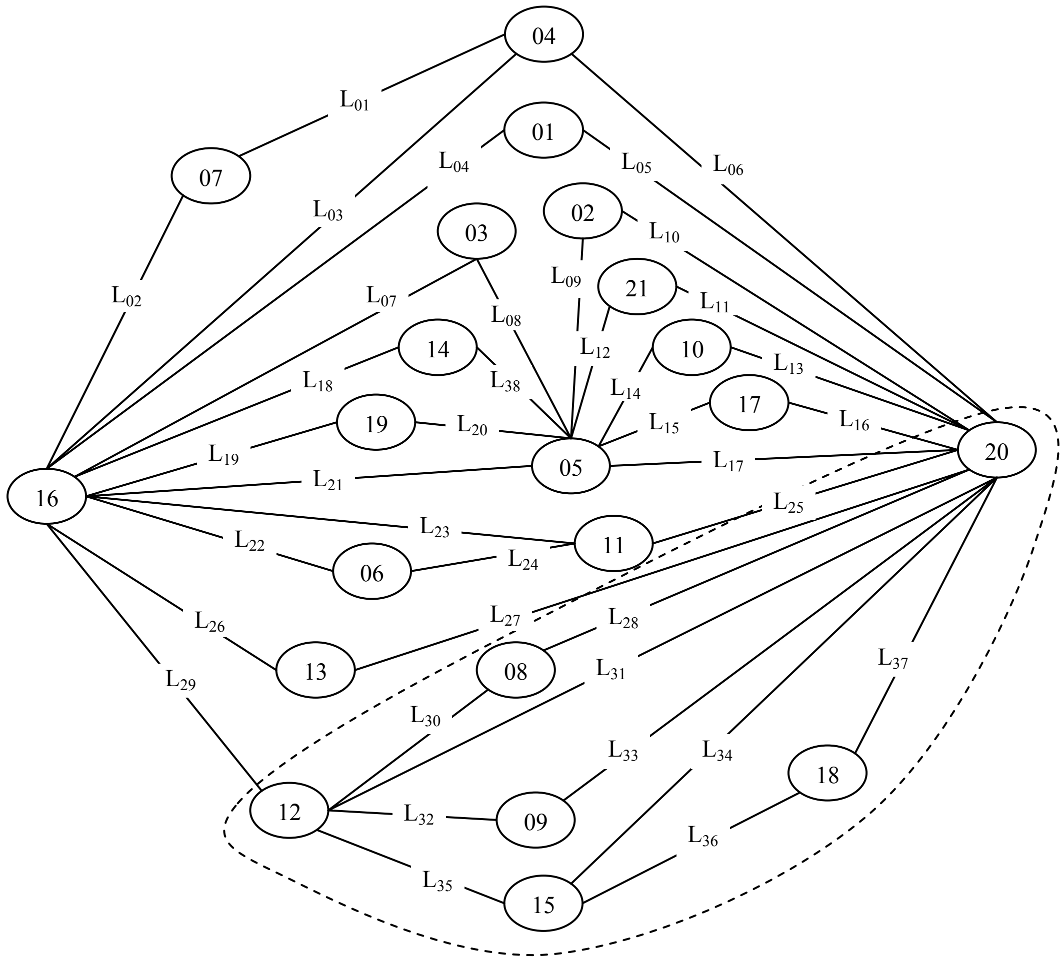



Figure 5.9 Liaison diagram of the drive assembly

The Figure 5.10, Figure 5.11 and Figure 5.11(a) shows below the front view, liaison graph model and directions of subassembly of an electromotor having the part numbers 08, 09, 12, and 20.

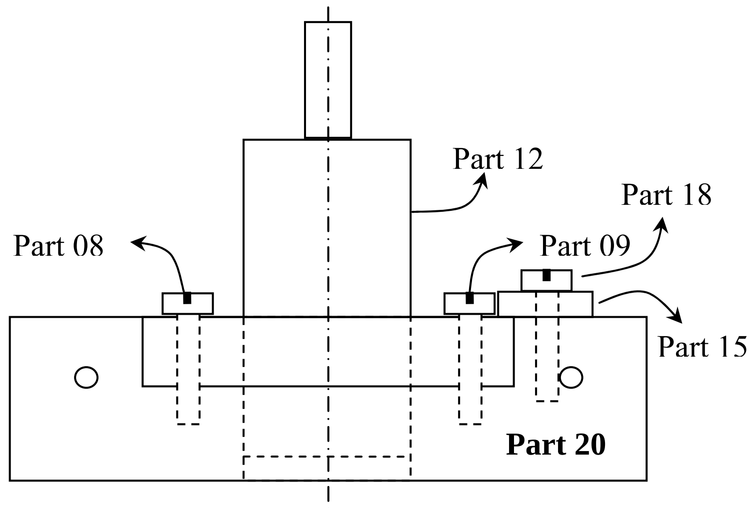



Figure 5.10 Front view of electromotor subassembly

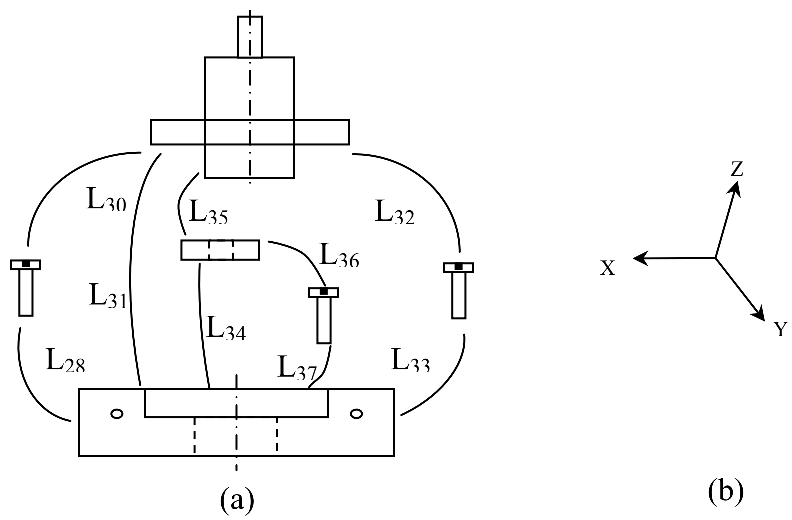



Figure 5.11 (a): Liaison graph model of electromotor subassembly; (b): Directions for assembly or disassembly

For the purpose of understanding the proposed methodology, let's consider three alternatively sequences of the drive assembly including one subassembly (shown within the dotted rectangle). These sequences are taken from the results of generation of assembly sequence using ACO.

Part Sequence 1:

$\boxed{20\text{-}12\text{-}15\text{-}18\text{-}8\text{-}9}$-4-11-13-1-5-17-2-21-10-16-14-3-7-6-19

Part Sequence 2:

$\boxed{20\text{-}12\text{-}15\text{-}18\text{-}8\text{-}9}$-4-11-13-1-16-7-6-5-17-2-14-3-21-10-19

Part Sequence 3:

$\boxed{20\text{-}12\text{-}8\text{-}9\text{-}15\text{-}18}$-4-11-13-1-16-7-6-5-17-2-14-3-21-10-19

Sequence 3 is arbitrarily chosen only for the sake of understanding the methodology. In practice, the optimization techniques are applied to determine the optimal sequence and then the same is picked for allocation to the robots. For the same problem of task assignment in multirobot environment has been conceived with twenty two task assemblies with two robots.

Task allocation to the available robots can be made using the three options.The allocations are made on the basis of capability of the robots so that the motion conditions are satisfied. For the purpose of a sample study, option-1 is used and the following allocations are made.

Task Allocation 1:

P→P→P→P→A→P→P→A→A→A→A→A→P→P→A→A→A→A→P→P→P→P

Task Allocation 2:

A→A→P→P→P→P→P→A→A→A→A→A→P→P→P→P→P→P→A→A→A→A

Task Allocation 3:

A→P→P→P→A→P→A→P→P→A→A→P→P→P→A→A→A→A→A→A→A→A

In the task allocation ‘A’ represents Adept One XL robot and ‘P’ represents Puma 560 robot. These three allocations are considered as candidates for optimization on the basis of a single objective called cyclic time of assembly.

### 5.7.3 Determination of cycle time

i) Allocation 1

a) Time taken by Puma 560

$12X + (\theta/1.138)10 = 12X + 8.78\,\theta$

The equivalent time ($\epsilon_i$) = A.X+0.5.B.R.X, where, A and B are the coefficient of X and θ respectively. R is the maximum reach of the robot arm. i = 1 for Puma and i = 2 for Adept.

Hence, $\epsilon_1 = 12X + 0.5\times 8.78\times 0.878X = 15.854X$

b) Time take by Adept one XL

$10(X/1.2) + (\theta\,/1.5)\,8 = 8.3X + 5.33\,\theta$

$\epsilon_2 = 8.3X + 0.5\times 5.33\times 0.8X = 10.432X$

Total equivalent cycle time taken by the robots is

$\epsilon = \epsilon_1 + \epsilon_2 = 15.854X + 10.432X = 26.286X$

ii) Allocation 2

a) Time taken by Puma 560

$11X + (\theta\,/1.138)\,8 = 11X + 7.02\,\theta$

$\epsilon_1 = 11X + 0.5\times 7.02\times 0.878X = 14.082X$

b) Time take by Adept one XL

$11(X/1.2) + (\theta\,/1.5)\,10 = 9.16X + 6.66\,\theta$

$\epsilon_1 = 9.16X + 0.5\times 6.66\times 0.8X = 11.824X$

$\epsilon = \epsilon_1 + \epsilon_2 = 25.906$

iii) Allocation 3

a) Time taken by Puma 560

$9X + (\theta\,/1.138)\,8 = 9X + 7.02\,\theta$

$\epsilon_1 = 9X + 0.5\times 7.02\times 0.878X = 12.082X$

b) Time take by Adept one XL

$13(X/1.2) + (\theta\,/1.5)\,9 = 10.83X + 6\,\theta$

$\epsilon_1 = 10.83X + 0.5\times 6\times 0.8X = 13.23X$

$\epsilon = \epsilon_1 + \epsilon_2 = 25.312X$

### 5.8 Summary

In this chapter, different methods of task allocation for MRS, based on six different methods viz. GH, LP, MILP, KA, HA and PSO. A comparative study vis-à-vis their applicability of these methods is also presented. Each method has been applied on same example problem to evaluate them on a common platform. In order to combine the processes and approach the problem in a holistic manner, a separate example of drive assembly is considered for task allocation. The results have been critically viewed for their suitability in the context of the present goal set and are discussed in Chapter 6.

# Chapter-VI

# RESULTS AND DISCUSSIONS

# CHAPTER 6

## Results and Discussions

### 6.1 Introduction

The strategies for robot selection, formation of multi-robot cells and operation of the same have been presented in chapter-3. The model for robot selection and those for task assignment under individualistic manner and in integrated manner are presented with all details in chapter-4 and chapter-5 respectively. The results obtained by using various models and methods for the MRS under consideration are presented in the following sections in the broad category of;

- Results on strategies for task allocation
- Results on robot selection
- Results on task assignment
- Results on integrated task assignment

### 6.2 Strategies for task allocation

An empirical study is described in the present work that sought general guidelines for task allocation strategies. Different task allocation strategies are identified, and demonstrated in the multi-robot environment. A simulation study of the methodology is carried out in a simulated grid world. The results show that there is no single strategy that produces best performance in all cases, and that the best task allocation strategy changes as a function of the noise in the system. This result is significant, and shows the need for further investigation of task allocation strategies.

### 6.2.1 Results of grid world frame work

A simplified version of the above described multi-robot task in a grid world is illustrated in Figure 6.1. As the base case of the grid world implementation, a 10×10 grid inhabited by 10 robots is considered. Robots bid on tasks depending on their capability (expressed by a number) to those tasks. The bid was set to 20- d, where d is the Manhattan distance to the task. In each time-step, any robot assigned to a particular task selects that task. When a robot selects a task, that task goes off the list and new tasks are added to it. In the context of emergency handling, commitment means that robots stay focused on a single task, until the task is over. The opposite, opportunism, means that robots can switch tasks, if for example another task is found with greater intensity or priority. In the experiments, coordination is linked to communication, namely the ability of robots to communicate about who should service which tasks, as opposed to individualism, where robots have no awareness of each other. Communication is used to prevent multiple robots from trying to accomplish the same task; robots inhibit others from engaging in the same task. The goal is to reduce interference among robots, and to prevent loss of coverage in some areas because all the robots rush to perform task in another area. Deciding the level of commitment and collaboration are key aspects of the multi-robot task allocation problem. The strategies are obtained by crossing individualism (I) and mutual exclusion (M) with opportunism (O) and commitment (C). Four alternatives were designed resulting from the combinations in varying the two parameters, coordination and commitment. The results of the grid world simulation are presented in Table 6.1. On one axis commitment versus opportunism is considered while on the other individualism versus mutual exclusion is considered.

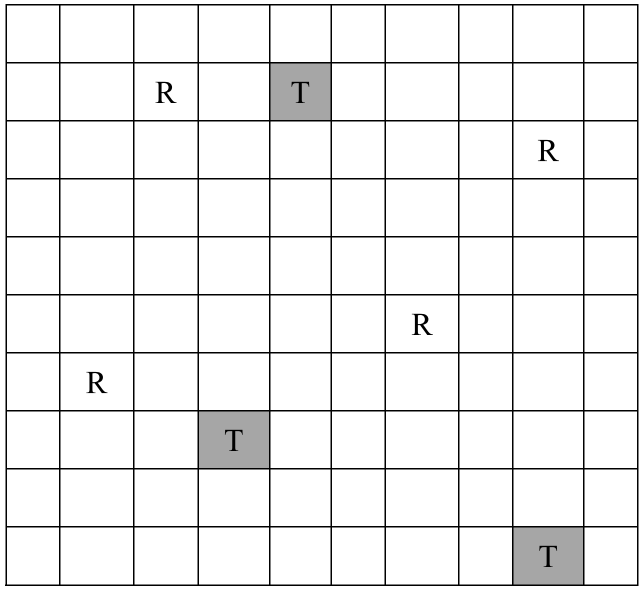



Figure 6.1 An example 10 x 10 grid world with four robots (R) and three tasks (T)

Table 6.1 Results from base case grid world

| Strategy: | I,O | I,C | M,O | M,C |
|---|---|---|---|---|
| Results: | 980 | 1045 | 435 | 722 |

### 6.2.2 Discussions of strategies for task allocation

The grid world results are interesting if they actually represent real world system behavior. The fact that the best performing task allocation strategy changes as to vary noise parameters in the grid world implies that it can be very difficult to decide *apriori* which task allocation strategy should be used in a given task for any real world implementation. The results clearly show that the opportunistic strategy worked significantly better than the commitment-based strategy. This might be because the time to reach a task was significantly larger than the time to complete a task, once a robot was there. This choice of parameters favors opportunism over commitment since the former effectively uses the presence of robots near emergencies by harnessing them immediately. In other regions of the parameter space of the emergency handling task (e.g., where the ratio of time-to-reach-task to time-to-complete-task is small) opportunism might not be as effective. The present study excluded the case where several robots would be required to do a task in a cooperative fashion, a regime in which performance might improve with commitment.

The selected four task allocation strategies are extreme, in that they take into consideration only the complete presence or absence of commitment and coordination in the given context. Arguably, the best strategy for any particular task would most likely be a carefully balanced compromise. However, as stated previously, the goal of this work was not to attempt to find the best strategy (which is necessarily task- and parameter-specific), but rather to gain some insight into task allocation in general. The four strategies explored provide a reasonable span of strategy space and provide leading insights for further study. In practice, the robot capability ratings can be obtained from the databases. Therefore, one can automatically select appropriate candidate for a given task by using the proposed matching procedure and databases.

### 6.3 Selection of robot

Essentially four different types of approaches are adopted in selection of robot. These are:

- In the first kind of approach, a methodology based on fittness methods is adopted which helps in selection of a suitable robot from among a large number of available alternative robots.
- The second approach is a capability based method and can consider any number of quantitative and qualitative robot selection attributes simultaneously and offers a more objective, simple and consistent robot selection approach.
- The third approach considers the robot parameters as well as the task parameters to form a model for relative ranking of the available candidate robots.

- The fouth method is a cases based method and it is different from existing commercial design systems. The method proposed can provide designers an advisory service based on previous experience.

The following sections present the results obtained through all these methods and the related discussions and comparisons.

### 6.3.1 Results of selection of robots by fittness

The weighted normalized attributes for the +ve and –ve benchmark robots are obtained through the methods described in section 4.4.1. The results are as follows;

$V^{*}$= (0.1129, 0.097, 0.064, 0.048, 0.074, 0.047, 0.246)

$V^{-}$ = (0.075, 0.032, 0.051, 0.025, 0.02, 0.015, 0.061)

The separation from the +ve and –ve benchmark robots are found as;

$$S_1^* = 0.044, S_2^* = 0.195, S_3^* = 0.096, S_4^* = 0.196$$

$$S_1^- = 0.206, S_2^- = 0.069, S_3^- = 0.185, S_4^- = 0.0083$$

The relative closeness values of the robots to the ideal solution are given in Table 6.2. The robots are ranked in order of preference based on the significant attributes chosen keeping the intended application in view. According to the results of the example problem, Robot-4 that has the lowest closeness value should be recommended as the best robot alternative. The 1$^{st}$ ranked robot has the highest DOF, cost, payload and swept area the best repeatability figures amongst all the robots. The 2$^{nd}$ ranked robot has the lowest cost, swept area, max.reach and repetability with highest pay load capacity. In order to discriminate between these two robot alternatives the closeness rating should be looked at. In the data set, P, G and VG denote poor, good, and very good respectively (Table 6.2). In order to determine the order of preference of the robot alternatives with respect to the closeness increase in throughput criterion, a ranking procedure becomes essential.

Table 6.2 Selection of robot

| Sl. No | Robot | Closeness | Ranking | Rating |
|---|---|---|---|---|
| 1 | Robot-1 | 0.824 | 4 | P |
| 2 | Robot-2 | 0.261 | 2 | VG |
| 3 | Robot-3 | 0.658 | 3 | G |
| 4 | Robot-4 | 0.04 | 1 | VG |

### 6.3.2 Results of selection of robots on the basis of capability

The relative ranking and ranking factors are calculated in section 4.4.2. The robots are arranged in order of their ranking factor based on the significant attributes chosen keeping the application of the robots in view. According to the results obtained and the analysis thereby, Robot-4 that has the lowest ranking factor should be recommended as the best robot alternative. The 1$^{st}$ ranked robot has the highest DOF, cost and payload. The 2$^{nd}$ ranked robot has the highest reach and repeatability. These robots are recommended for selection for performing the intended tasks. As a result of the application of both numerical and qualitative inputs and outputs, two robot alternatives are found to be more efficient compared to other candidates. In order to discriminate between these two robot alternatives the ranking factor should be looked at. The ranking curves of robots are shown in Figure 6.2. The average values of the ranking factor are presented in Figure 6.3.

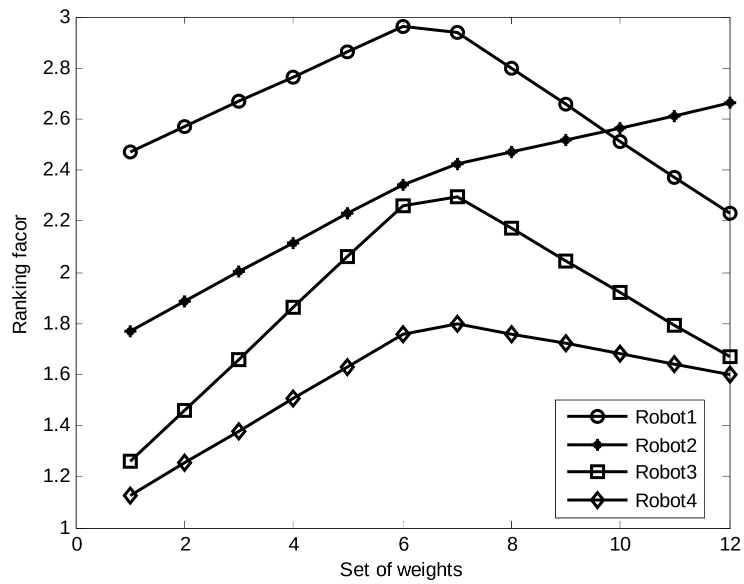



Figure 6.2 Ranking curves of robots

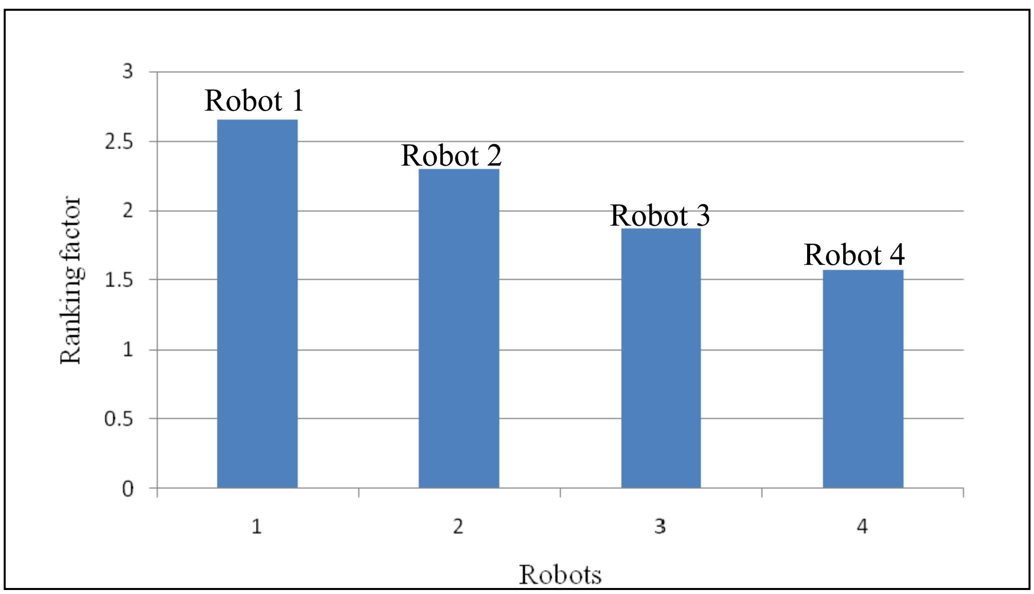



Figure 6.3 Comparison of robots

On the basis of the ranking factors the robots are rated as ‘Low’ or ‘High’ in relation to the group of the robots under consideration and are shown in Table 6.3. The procedure provides a coding system for robots depicting the various attributes. It recognizes the need for, and processes the information about, relative importance of attributes for a given application without which inter-attribute comparison is not possible.

In this work also as many as 30 attributes of the robots are recognized and codify successfully. The methodology developed through this work would help production engineers to select robots for the intended application.

Table 6.3 Scores of robot

| Sl. No | Robot | Ranking factor | Relative ranking | Relative rating |
|---|---|---|---|---|
| 1 | Robot-1 | 2.65 | 4 | Low |
| 2 | Robot-2 | 2.3 | 3 | Low |
| 3 | Robot-3 | 1.871 | 2 | High |
| 4 | Robot-4 | 1.571 | 1 | High |

### 6.3.3 Results of selection of robots on the basis of task requirement

The robots are arranged in order of their ranking factor based on the significant attributes chosen keeping the application of the robots in view. The details of calculations are explained in section 4.4.3. According to the results obtained and the analysis thereby, Robot-5 and Robot-4 has the highest ranking factors should be recommended as the best robot alternative. The 1$^{st}$ and 2$^{nd}$ ranked robots have the highest figures amongst all the robots. As a result, two robot alternatives are found to be more competent compared to other robots. The ranking curves of robots are shown in Figure 6.4. The average values of the ranking factor are presented in Figure 6.5.

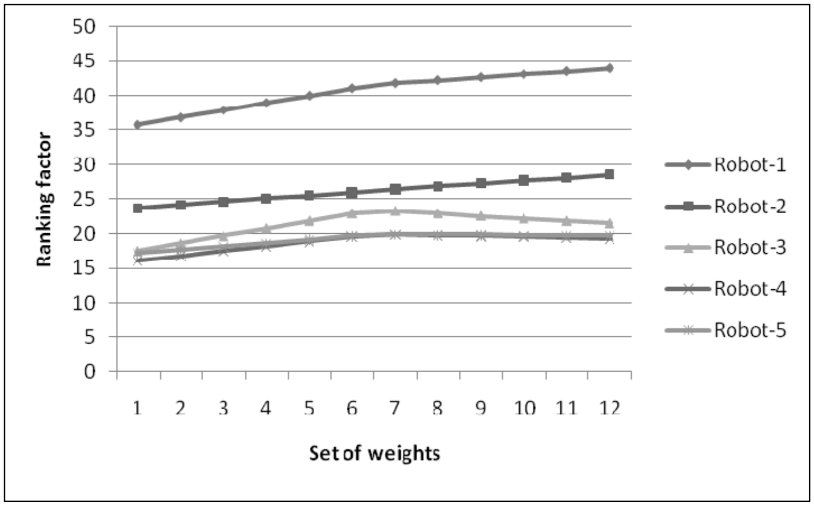



Figure 6.4 Ranking curves of robots

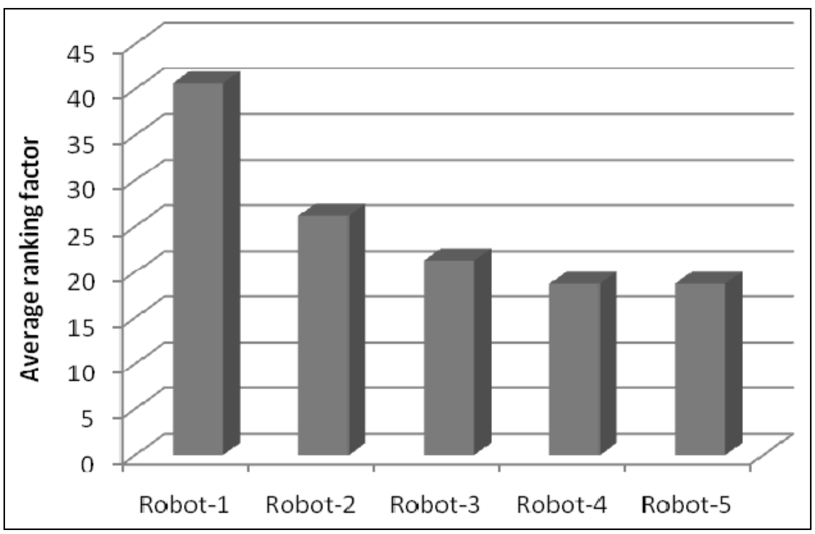


Figure 6.5 Comparison of robots

Although ranking of robots on the basis of the manipulators parameters alone has been attempted by some previous researchers, ranking of the robots in view of performing a given set of tasks is a novel attept. The values of these ranking factors for all the robots are given Table 6.4. On the basis of the ranking factors the scores of robots are rated as 'Low', 'Medium' and 'High' and in relation to the group of the robots under consideration and are shown in Table 6.5. The procedure provides a coding system for robots depicting the various attributes. It recognizes the need for, and processes the information about, relative importance of attributes for a given application without which inter-attribute comparison is not possible. Essentially the present work contributes to developing a methodology based on matrix methods which helps in selection of a suitable robot from among a large number of available alternative robots.

Table 6.4 Values of total ranking factor

| Value of $\sum \sigma$ with different set of weights | | | | | |
|---|---|---|---|---|---|
| Weight/ Robot | Robot-1 | Robot-2 | Robot-3 | Robot-4 | Robot-5 |
| $W_1$ | 35.7355 | 23.7325 | 17.5315 | 16.1235 | 17.2025 |
| $W_2$ | 36.799 | 24.1855 | 18.6095 | 16.8095 | 17.688 |
| $W_3$ | 37.8625 | 24.6385 | 19.6875 | 17.4955 | 18.1735 |
| $W_4$ | 38.926 | 25.0915 | 20.7655 | 18.1815 | 18.659 |
| $W_5$ | 39.9895 | 25.5445 | 21.8435 | 18.8675 | 19.1445 |
| $W_6$ | 41.053 | 25.9975 | 22.9215 | 19.5535 | 19.63 |
| $W_7$ | 41.798 | 26.4325 | 23.2805 | 19.8415 | 19.857 |
| $W_8$ | 42.2245 | 26.8495 | 22.9205 | 19.7315 | 19.8255 |
| $W_9$ | 42.651 | 27.2665 | 22.5605 | 19.6215 | 19.794 |
| $W_{10}$ | 43.0775 | 27.6835 | 22.2005 | 19.5215 | 19.7625 |
| $W_{11}$ | 43.504 | 28.1005 | 21.8405 | 19.4015 | 19.731 |
| $W_{12}$ | 43.9305 | 28.5175 | 21.4505 | 19.2915 | 19.6995 |

Table 6.5 Scores of robot

| Sl. No | Robot | Average ranking factor | Relative ranking | Relative rating |
|---|---|---|---|---|
| 1 | Robot-1 | 40.6293 | 4 | Low |
| 2 | Robot-2 | 26.17 | 3 | Medium |
| 3 | Robot-3 | 21.304 | 2 | Medium |
| 4 | Robot-4 | 18.703 | 1 | High |
| 5 | Robot-5 | 18.703 | 1 | High |

### 6.3.4 Results of selection of robots using case based approach

The detailed calculations for selection of robots are explained in section 4.4.4. In the query problem, a user is requested to fill up the required data such as price, reach, production rate, and work envelop requested by the system. After the user has specified the input requirements, they can use the search function to find out 10 of the most similar cases to generate final solutions. The efficiency of the system is primarily related with the representation of the query problem. The answer of the query problem cannot be retrieved until each matching case is analyzed, mapped, and transferred. In this process, reuse of case solution not only increases the efficiency, but also improves the quality of solving new problems.

Table 6.6 Comparision of robots with the standard one

| Features | Query problem | Robot -1 | Robot -2 | Robot -3 | Robot - 4 | Robot -5 |
|---|---|---|---|---|---|---|
| Similarity (%) | 100 | 81 | 84.5 | 83.6 | 79.7 | 83.2 |
| Price (In 1000 US$) | 100 | 105 | 103 | 105 | 105 | 103 |
| Repetability (mm) | ± 2 | ± 1 | ± 1 | ± 1 | ± 1 | ± 1 |
| Reach (mm) | 2600 | 2400 | 2200 | 2400 | 2200 | 2200 |
| Payload (kg) | 100 | 120 | 120 | 120 | 120 | 120 |
| Velocity (mm/s) | 5000 | 5500 | 5400 | 5000 | 5400 | 5500 |
| D.O.F | 5 | 6 | 6 | 5 | 5 | 5 |
| Geometry | Cartesian | Articulated | Articulated | Sphrical | Articulated | Cartesian |

After having completed the initial retrieval, the best matching case is presented in front of the user in Table 6.6. Then, users have to select one of the optimal cases to adapt and refine the solution. In the step of selecting the optimal cases, many similarity features (or attributes) are employed to calculate the final similarity. Because each feature of the case has different effects on solving the query problem, and assign the weight to each feature as well. In our example, confirmed case 2 is the most similar to the query problem with final similarity value 84.5%. In order to improve the quality of the selecting process, retrieval mechanisms and case representations have become a major topic to increase system performance. The case of similarity between robot features is shown in Figure 6.6.

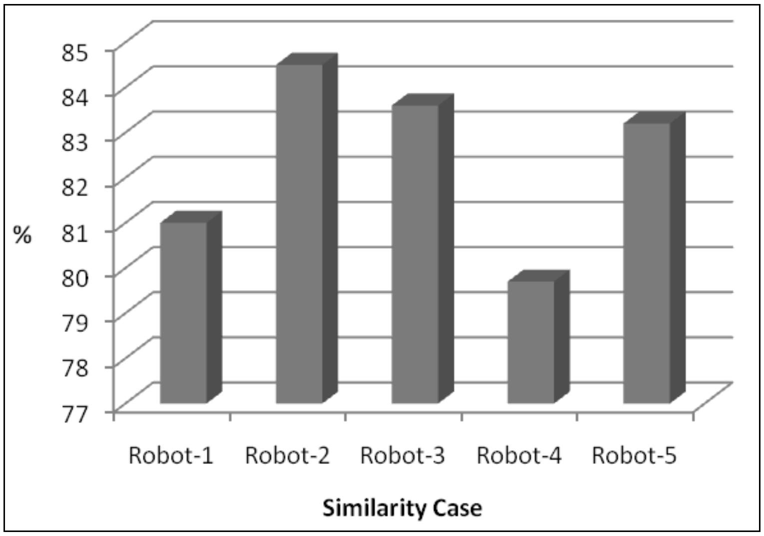



Figure 6.6 Comparison of robots as per similarity case

### 6.3.5 Discussions

The fittness based method presents a robot selection procedure based on the multiple attribute based approach, which is a concept used not so frequently for this purpose. It identifies the various attributes needing to be considered for the optimum evaluation and selection of robots. A robot alternative with a relatively low closeness is more likely to exhibit good performance.

Robot selection is a multi-attribute decision making process and the result can provide an optimum solution for selecting candidate robots in the capability based method. On the basis of the relative ranking factors the selection robots are finalized.

The robot selection on the basis of task requirement is aimed at developng a generalized tool to combine manipulator attributes and task requirements in a comprehensive manner for relative ranking of the manipulators. In the intial phase of the formulation, 35 attributes of the robots are identified and consciously coded to take care of the characterstics of a robot manipulator precisely. The methodology developed through this work can be applied to any similar set-up .This is sure to help the designers and users in selecting the robots correctly for the intended application.

There have been many studies on production process and robot selection. The case based methodology is useful for selecting robots and workcell design. The technique presented is useful in reducing the chance of poor quality design, inexperienced mistakes, and long development lead time. As mentioned previously, selecting the right robots for workcell design may not be a simple and obvious task. The methodology is to help those inexperienced designers, as it bridges the gaps between product design and manufacturing stages and leaves no room for misconceptions and a poor foundation for design decision making.

### 6.4 Task allocation

The results obtained by using different methods for the allocation of MRS under consideration are presented in the following sections. As presented in chapter-5, six different types of allocation methods are adopted in the MRS. These are:

- Greedy Heuristic (GH) which help allocate the task to robot and is handled to obtain a feasible solution.
- Linear Programming (LP) based method. The technique is quite suitable and efficient for problems with limited number of tasks.
- Mixed Integrated Linear Programming (MILP) that yields optimized multirobot task allocation. This approach can be advantageously used in real-world problems.
- Knapsack Algorithm (KA) that can be advantageously used for problems of large size. The method proposed can provide designers an advisory service based on previous experience.
- Hungarian Algorithm (HA) for task allocation and the solutions obtained from this algorithm are feasible and the assignment of task to robot is uniformly distribured.

- Particle Swarm Optimization (PSO) is one of the latest evolutionary optimization techniques for MRS. Furthermore, PSO algorithm works well on most global optimal problems.
- An integrated approach for assembly sequence generation and task allocation for MRS by considering their capability in terms of time and space.

For all test problems excluding the integrated approach, four robot types with fixed charges are considered as candidate robots. The test problems are created by a problem generator using four major design parameters. These are:

i) The average robot service capacity (i.e., the average number of workstations that can be served by a robot based on one-dimensional resource demand of workstations);

ii) The average space required by the given workstations;

iii) The average machine time required by the given workstations;

iv) The number of workstations to be assigned.

For the integrated approach, an example of a 21-part drive assembly with 2 robots is taken for the task allocation. The following sections present the results obtained from the aforementioned task allocation methods and the allied discussions and comparisons.

### 6.4.1 Results of Greedy Heuristic for assignment

The robustness and effectiveness of the optimization algorithms are examined by generating problems and testing them based on the key design parameters. The first optimization algorithm was coded in MATLAB for solving the greedy heuristic problems. GH is usually faster, since they don't consider the details of possible alternatives. An algorithm that always takes the best immediate or local solution has the possibility of getting trapped locally. Hence, greedy algorithms find the overall or globally optimal solution for some optimization problems, but may find less-than-optimal solutions for some instances.

The results of the allocation obtained from the GH approach are presented in Table 6.7. Since macro planning for a multi-robot system is quite important to a designer, the onetime computing cost for optimization should not be a major concern. Considerable and valuable results are developed in GH. The total cost of assigned task is 14.563 as per GH solution. The utilizatios of GH is shown in Figure 6.7.

Table 6.7 Task assignment using GH

| Robot | Assigned workstation |
|---|---|
| Robot-1 | Task-1 |
| Robot-2 | Task -2, Task -3, Task -4 |
| Robot-3 | Task -5, Task -6, Task -7, Task -8, Task -9 |
| Robot-4 | Task -10, Task -11, Task -12, Task -13, Task -14, Task -15 |

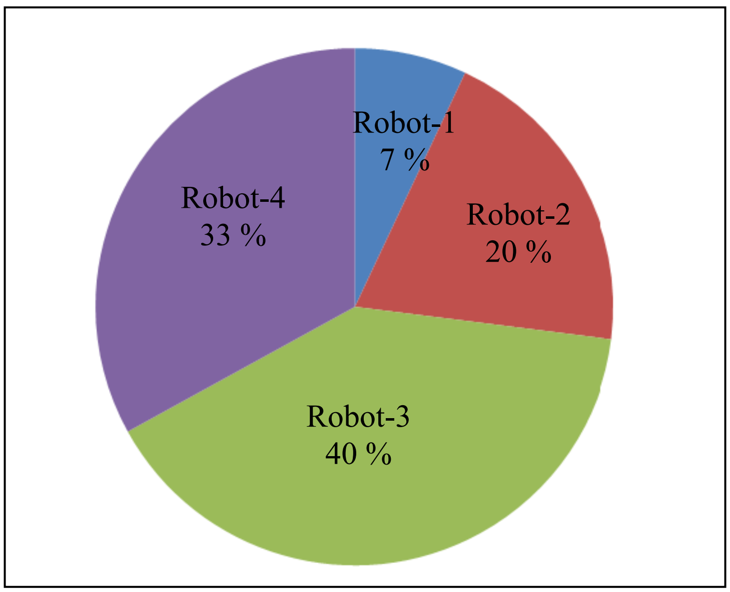



Figure 6.7 Utilization of robots using GH

### 6.4.2 Results of Linear Programming for assignment

The second optimization algorithm based on LP was coded in LINGO. The total cost of assigned task is 13.931. The results of the allocation are presented in Table 6.8. Overall, the computational results indicate the initial feasible solution generated by the LINGO takes no more than a second. The quality of the solution is reasonably good. The solution times for finding a near-optimum or an optimum are also recorded. Thus, the algorithms developed in this work provide significant and useful results. The result also implies that the size of the LP problem is determined by the number of tasks, and is independent of the number of robots.

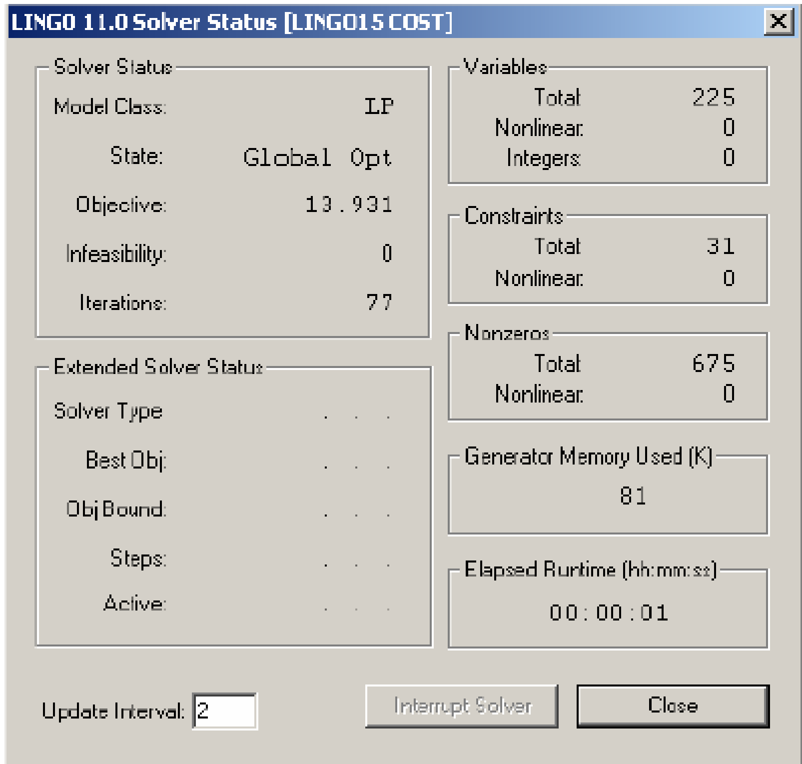



Figure 6.8 Results of the LP using LINGO

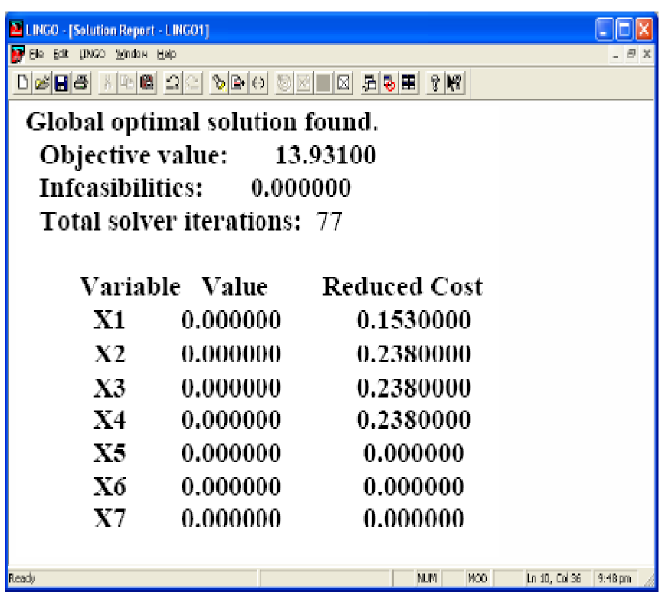



Figure 6.9 Solution report of the LP using LINGO

Table 6.8 Task assignment using LP

| Robot | Assigned workstation |
|---|---|
| Robot-1 | Task -13 |
| Robot-2 | Task -12, Task -14, Task -15, |
| Robot-3 | Task -5, Task -6, Task -7, Task -8, Task -10, Task -11 |
| Robot-4 | Task -1, Task -2, Task -3, Task -4, Task -9 |

The Solver Status box as shown in Figure 6.8 details the model classification (LP, QP, ILP, IQP, NLP, etc.), state of the current solution (local or global optimum, feasible or infeasible, etc.), the value of the objective function, the infeasibility of the model (amount constraints are violated by), and the number of iterations required to solve the model. After the solver status box the LINGO displays a solution report regarding the values of each variable and the complete allocation that will produce the optimal value of the objective function. The reduced cost for any variable that is included in the optimal solution is always zero. For variables not included in the optimal solution, the reduced cost shows how much the value of the objective function would decrease (for a MAX problem) or increase (for a MIN problem) if one unit of that variable were to be included in the solution. The solution report of the LP using Lingo with the detailed assignment with the optimized value is shown in Figure 6.9. The utilization of robots using LP is shown in Figure 6.10.

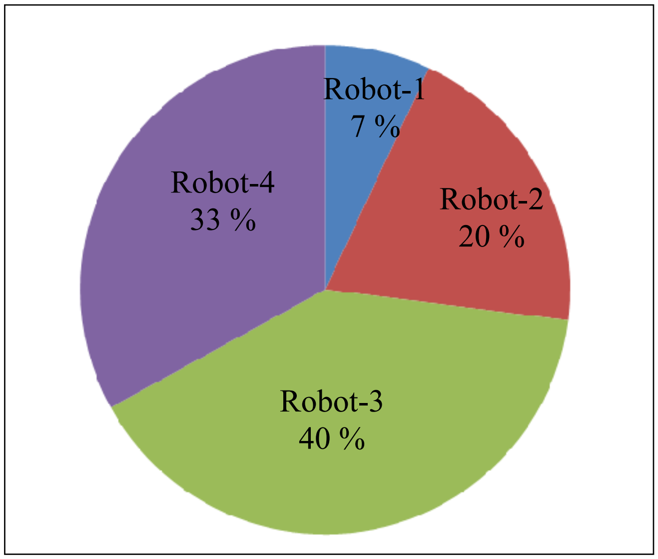



Figure 6.10 Utilization of robots using LP

### 6.4.3 Results of MILP for assignment

The results of allocation obtained from the MILP are presented in Figure 6.11. Due to the restrictions of Management Scientist software 10 tasks with four robots are solved and are compared with all methods with 10 tasks and it is to be found that it is suitable for practical size problems. It may be mentioned that MILP problems can also be solved using LINGO but the third optimization algorithm was coded in Management Scientist for solving the MILP for the sake of testing the effectiveness of the tool. The comparison of run time between LINGO and Management Scientist is checked and it is observed that both the tools are equally effective for the problem.

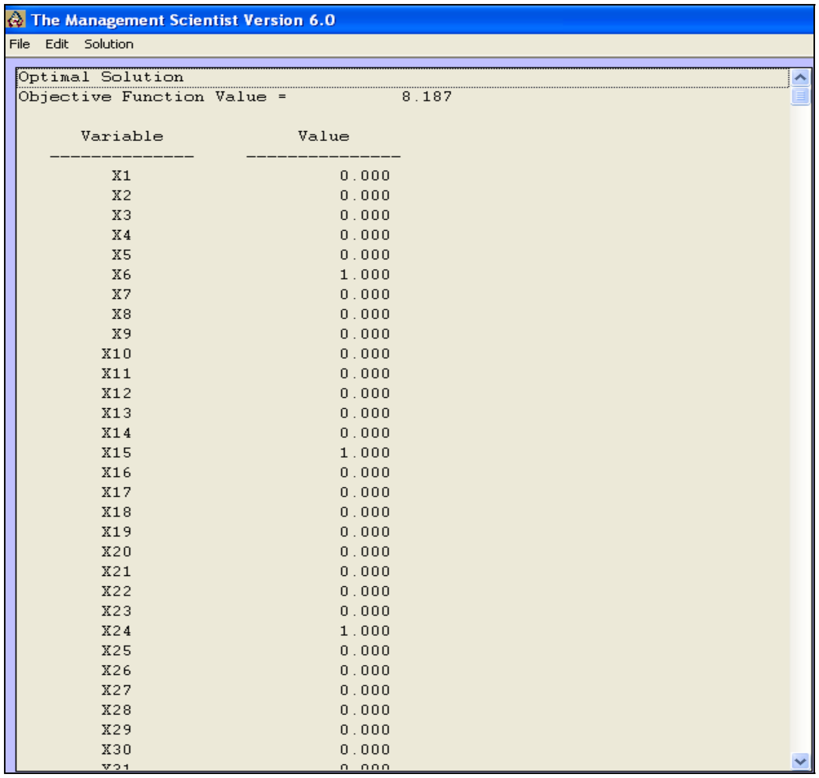



Figure 6.11 Results of the MILP using Management Scientist

Therefore, while the MILP can model the most complex classifications of task allocation problems, the solutions are limited to instances where there is small number of robots or tasks. By modeling the MILP problem using Management Scientist optimization software, one can gather data about the complexity of different problem instances and determine the limits to the problem size that the simulation can feasibly handle.The rate of complexity grows as more variables are added. The total cost of assigned task is determined to be 8.714.

Table 6.9 Task assignment using MILP

| Robot | Assigned workstation |
|---|---|
| Robot-1 | Task-8 |
| Robot-2 | Task-9, Task-10 |
| Robot-3 | Task-4, Task-5, Task-6, Task-7 |
| Robot-4 | Task-1, Task-2, Task-3 |

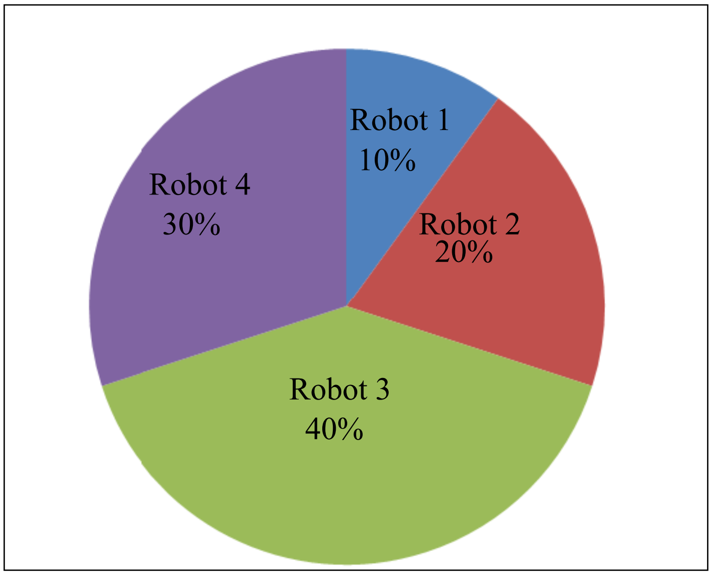



Figure 6.12 Utilization of robots using MILP

The outcome of the method of allocation in terms of the robots utilization is presented in Figure 6.12. The initial feasible solution generated by the heuristic takes no more than a second. The solution times for finding a near-optimum or an optimum are also recorded. The results of the allocation are presented in Table 6.9.

### 6.4.4 Results of Knapsack Algorithm for assignment

The fourth optimization algorithm based on KA was coded in LINGO. The results of the allocation are presented in Table 6.10. Furthermore, it is possible to use KA for solving the large scale knapsacks, as it is independent of robots as well as number of tasks. The total cost of assigned task for the present example problems is found to be 13.794. Solution report and utilization of the results of allocation are obtained from this method is presented in Figure 6.13 and Figure 6.14, where from it is evident that the allocation cost is lower in KA compared to LP, MILP and GH, KA is more capable for large scale problems.

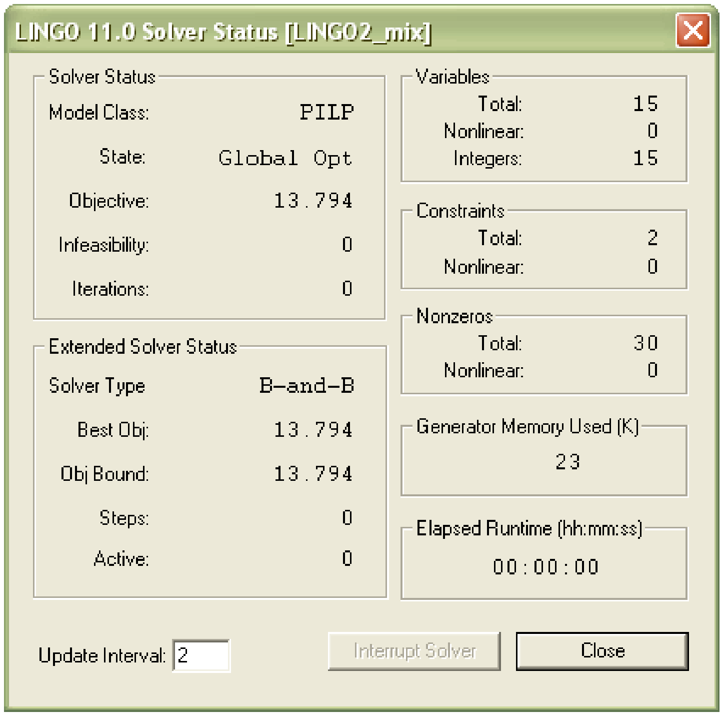



Figure 6.13 Results of the KA using LINGO

Table 6.10 Task assignment using KA

| Robot | Assigned workstation |
|---|---|
| Robot-1 | Task-4, Task-5, Task-6 |
| Robot-2 | Task-1, Task-2, Task-3 |
| Robot-3 | Task-7, Task-8, Task-9, Task-13, Task-14 |
| Robot-4 | Task-10, Task-11, Task-12, Task-15 |

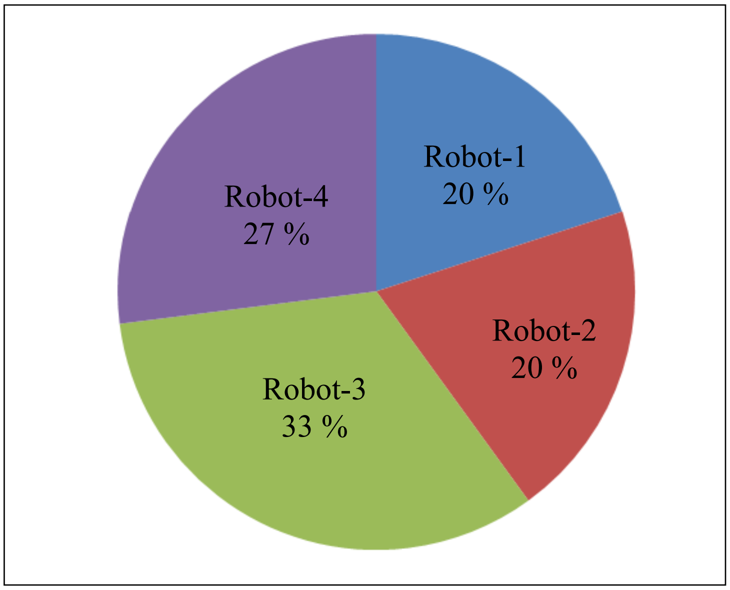



Figure 6.14 Utilization of robots in KA

In an MRS typically multiple robots are available to accomplish large number of tasks. Any robot can be assigned to perform any task, incurring some cost that may vary depending on the robot-task combination. Large multiple knapsack problems, despite the NP-hardness, generally are as easy to solve as ordinary 0-1 knapsack problems.

### 6.4.5 Results of Hungarian Algorithm for assignment

The fifth optimization algorithm based on HA was coded in Matlab. The total cost of assigned task is 15.032. From the Matlab program it is observed that the runtime of HA is very small. HA will iterate several times until a feasible schedule is obtained where all tasks are assigned to robots. After every iteration the resulting assignments are checked for the overlapping and conflict of tasks in terms of robots' schedules (two robots must not be scheduled in the same time where a task must attend both). The results of the allocation are presented in Table 6.11. The outcome of the method of allocation in terms of the robots utilization is presented in Figure 6.15.

Table 6.11 Task assignment using HA

| Robot | Assigned workstation |
|---|---|
| Robot-1 | Task-4 |
| Robot-2 | Task-1, Task-2, Task-3 |
| Robot-3 | Task -5, Task -6, Task -7, Task -8,Task -9, Task -10 |
| Robot-4 | Task -11, Task -12, Task -13, Task -14, Task -15 |

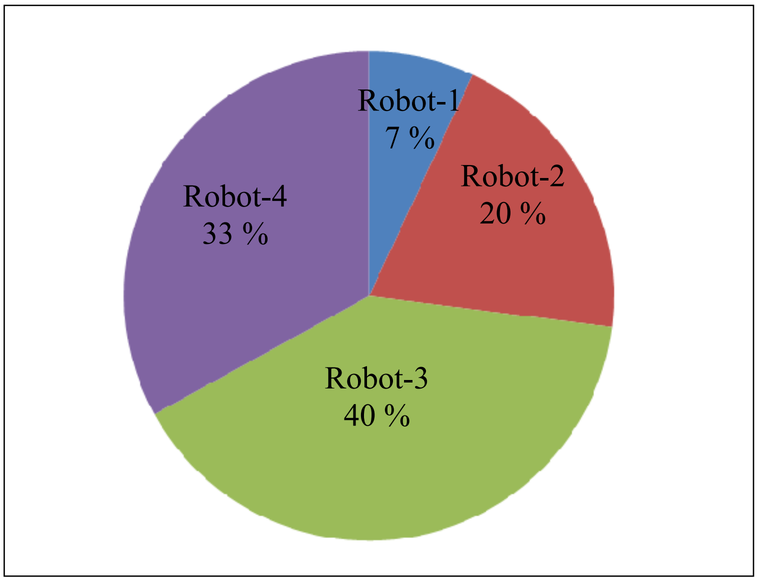



Figure 6.15 Utilization of robots in HA

### 6.4.6 Results of particle swarm optimization for assignment

The sixth optimization algorithm based on PSO was coded in Matlab. The total cost of assigned task is 13.674. It has the better ability of global searching and has been successfully applied to many areas. PSO algorithm is employed to solve the MRTA problem in a dynamic environment. The results show that PSO algorithm is effective for task allocation problems.This approach aims to generate an optimal schedule so as to get the minimum completion time while completing the tasks. PSO algorithm is an adaptive method that can be used to solve optimization problem. The task assignment and the robot utilizaions using PSO are shown in Table 6.12 and Figure 6.16.

Table 6.12 Task assignment using PSO

| Robot | Assigned workstation |
|---|---|
| Robot-1 | Task-13 |
| Robot-2 | Task-10, Task-12, Task-15 |
| Robot-3 | Task -1, Task -2, Task -5, Task -6,Task -7, Task -8 |
| Robot-4 | Task -3, Task -4, Task -9, Task -11, Task -14 |

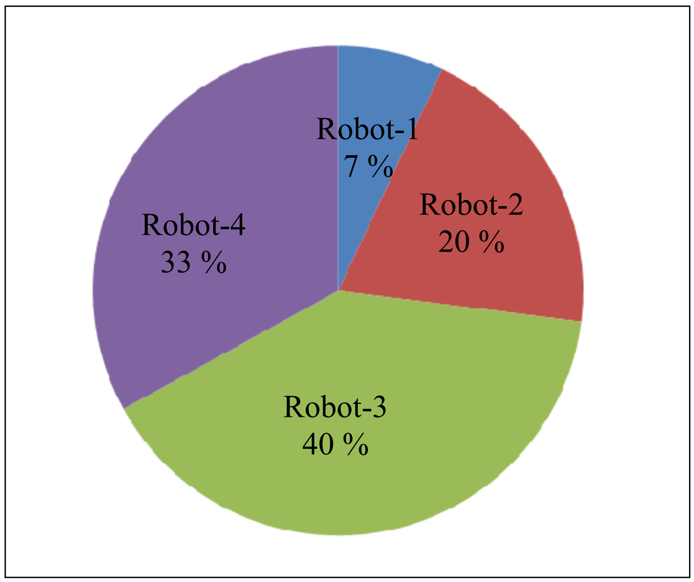



Figure 6.16 Utilization of robots in PSO

### 6.4.7 Discussions

The outcome implies that the size of the GH is independent to the number of tasks, and independent of the number of robots. GH problems can be successfully implemented in large size problems.

The utilization of robots are not taken care of properly in LP. The LP problems are only suitable for practical size problems. The task assignment is well distributed in LP.

The MILP provides significant and useful results. The MILP is also not suitable for larze size problems. The quality of the solution is reasonably good.

All the robots are utilized decently in the KA. As per utilization concerned, KA is the best methods as compared to the other methods. The KA achieves a good efficiency which is independent on the number of task and robots.

The results show that HA is able to be applied to the MRS effectively and it can satisfy the optimal need. It was observed that the runtime of HA is very small, but its allocation cost is high as compared to other methods.

PSO problems are well suited to larger size problems. The simulation of the program shows that it achieves the global solutions in a fraction of second.

### 6.5 Results of integrated method for MRTA

In order to test the developed methodology for practical problems and treat the methods in a holistic manner an example problem is considered wherein the integrated method of task division, robot selection and task assignment is followed. The problem is conceived with two industrial robots of different configurations as mentioned in the previous section. There are two sets of results for any assembly problem under consideration. The first set gives the alternate feasible and stable sequence of assembly while the second set gives the optimized cycle time for a production of a single product.

#### 6.5.1 Calculation of cycle time

*i) Allocation 1*

a) Time taken by Puma 560

$12X + (\theta/1.138)10 = 12X + 8.78\,\theta$

The equivalent time ($\epsilon_i$) = $A.X+0.5.B.R_i.X$, where, A and B are the coefficient of X and $\theta$ respectively. $R_i$ is the maximum reach of the robot arm. i = 1 for Puma and i = 2 for Adept.

Hence, $\epsilon_1 = 12X + 0.5\times 8.78\times 0.878X = 15.854X$

b) Time take by Adept one XL

$10(X/1.2) + (\theta/1.5)\,8 = 8.3X + 5.33\,\theta$

$\epsilon_2 = 8.3X + 0.5 \times 5.33 \times 0.8X = 10.432X$

Total equivalent cycle time taken by the robots is

$\epsilon = \epsilon_1 + \epsilon_2 = 15.854X + 10.432X = 26.286X$

*ii) Allocation 2*

a) Time taken by Puma 560

$11X + (\theta /1.138)\, 8 = 11X + 7.02\, \theta$

$\epsilon_1 = 11X + 0.5 \times 7.02 \times 0.878X = 14.082X$

b) Time take by Adept one XL

$11(X/1.2) + (\theta /1.5)\, 10 = 9.16X + 6.66\, \theta$

$\epsilon_1 = 9.16X + 0.5 \times 6.66 \times 0.8X = 11.824X$

$\epsilon = \epsilon_1 + \epsilon_2 = 25.906$

*iii) Allocation 3*

a) Time taken by Puma 560

$9X + (\theta /1.138)\, 8 = 9X + 7.02\, \theta$

$\epsilon_1 = 9X + 0.5 \times 7.02 \times 0.878X = 12.082X$

b) Time take by Adept one XL

$13(X/1.2) + (\theta /1.5)\, 9 = 10.83X + 6\, \theta$

$\epsilon_1 = 10.83X + 0.5 \times 6 \times 0.8X = 13.23X$

$\epsilon = \epsilon_1 + \epsilon_2 = 25.312X$

Table 6.13 Summary of cycle time

| Option No. | X | | θ | | Є |
|---|---|---|---|---|---|
| | P | A | P | A | |
| 1 | 12 | 8.3 | 8.78 | 5.33 | 26.286 |
| 2 | 11 | 9.16 | 7.02 | 6.66 | 25.906 |
| 3 | 9 | 10.83 | 7.02 | 6 | 25.312 |

It can be observed from Table 6.13 that, the option no-3 for allocation of tasks to the available robots yields the minimum cycle time with an equivalent total time of 25.312. This option is optimized with satisfying all the criteria mentioned in the formulation of allocation model. The quality of the solution is reasonably good. As noticed, the computing efficiency is very sensitive to the problem size. Thus, the technique developed in this work provides significant and useful results. The technique has been tested in a variety of situations, considering different product structures (number of parts, number of connections between parts), different type of optimized assembly sequences, and different assembly resources (number of robots and its specifications). The number of robots can be increased depending on the number of parts and their manipulation requirements. From the capacity and DOF, there are number of sequences to complete the task by the robots. In the example, since option-3 for task allocation has produced optimized time, the same is accepted for allocation to the robots. The results of the allocation are presented in Table 6.14.

Table 6.14 Task assignment using integration model

| Robot | Assigned workstation |
|---|---|
| Robot-1 (PUMA 560) | Task-18, Task-4, Task-11,,Task-6, Task-16, Task-7, Task-6, Task-12, Task-8, Task-9 |
| Robot-2 (Adept one XL) | Task-20-A, Task-15, Task-13, Task-1, Task-5, Task-17, Task-2, Task-14, Task-3, Task-21, Task-10, Task-19, Task-20, |

### 6.5.2 Discussions

The present work is an integrated approach towards designing an effective robotic assembly system environment for industries. The advanced manufacturing technology of today requires the use flexible devices for becoming more agile and competitive. An approach has been made through this work to plan for an effective, cost efficient and assembly system with minimum cycle time. The benefits such as ability to perform complex operations simultaneously by several arms increase of work cell reliability by sharing of responsibilities, reduction in space by space-sharing, increase in system flexibility by shared manufacturing and material handling resources, and reduction in cycle time by concurrent work amply justifies the use of multirobot systems in industries.

## 6.6 Observations

However, before a final decision is taken, the factors such as economic considerations, availability, management constraints and corporate policies, international market policies etc. may be considered. As many as 30 attributes of the robots are identified and an attempt has been made to codify most of the robot characteristics, which will define the robot precisely and accurately. The coding scheme is illustrated with example. The methodology developed through this work would help production engineers to select robots for robotic workcell design.

Diverse approaches have been proposed in the past decade to deal with industrial robot selection. In general, it is assumed that engineering attributes are mutually independent; however, this is a very critical assumption and might result in the selection of a robot alternative.

It also provides a coding system for robots depicting the various attributes. It recognizes the need for, and processes the information about, relative importance of attributes for a given application without which inter-attribute comparison is not possible. It presents the result of the information processing in terms of a merit value, which is used to rank the robots in the order of their suitability for the given application. The contributions of this work can be summarized as;

1. The method is especially suitable for generating database of robots available in the market and their subsequent retrieval. It provides coding scheme to produce electronic database of globally available robots.
2. This database will be helpful to all sorts of people related to robots from manufacturer, designers, and users to maintenance personnel. It will be helpful to improve the overall productivity of the organization.
3. Here by identifying 35 attributes of the robots, the attempt has been made to codify most of the robot characteristics, which will define the robot precisely and accurately. The coding scheme is illustrated with examples.
4. Evaluation and ranking based on the mathematical approaches along with the illustrative examples are given.

The robot selection on the basis of task requirement is aimed at developng a generalized tool to combine manipulator attributes and task requirements in a comprehensive manner for relative ranking of the manipulators. The methodology developed through this work can be applied to any similar set-up .This is sure to help the designers and users in selecting the robots correctly for the intended application. In this example a total of 13 parameters are taken into account. The factors such as arm geometry, actuator, control mode, robot programming, space and time are to be considered in this example for better selection of robots. As per the analysis, robot selection on the basis of task requirement is a best method for robot selection as compared to the other methods.

The case based method has presented a method for helping production engineers to select robots for robotic workcell design. On the basis of these example results, it may be concluded that this method is a suitable system as a robot selection application. When the cases increase, the system will become more useful for robot selection. In order to improve the performance, it is recommended that the system user should collect more cases from historical robot applications and production processes. The need to combine the product development activities with production process and robot selection information has been emphasized for many years.

Multirobot facility design and planning have become increasingly important in modem production over the past decade. In this work, a mathematical model and solution algorithm is developed to support robot selection and task assignment in a system employing multiple robot types. Specifically, our model considers selection of a proper mix of multipletype robots such that operational requirements for a given number of tasks are satisfied. Each robot is characterized by its unique fixed charge and subject to its machine time and space capacity constraints. Each task has known time and space demands for each type of robot.

An optimization algorithm is developed using a greedy heuristic. The model is formulated as a pure 0-1 mathematical program, which is shown harder than the two-dimensional bin packing problem, a well-known NP-hard problem. Computational results indicate that the algorithm is effective and efficient in solving problems of a practical size. The algorithm serves as a practical tool for planning facilities with multiple types of robots.

A model of LP is developed for multirobot assignment. The result implies that that the size of the linear programming is determined by the number of tasks, independent of robots linear. The model is initially formulated as a pure 0-1 mathematical program. The initial solution obtained from the first phase is utilized to decide the task performing capacities of the candidate robots. The model is then simulated by number of tasks to make it suitable for application of LP in order to find out the optimized task allocation. In order to test the efficiency of the methodology an example problem with four heterogeneous robots and fifteen different tasks is worked out. Computational results indicate that the algorithm is effective and efficient in solving problems of a practical size.

A mathematical model and solution algorithms are developed to support robot selection and task assignment in a system employing multiple robot types. Models of MILP are developed to solve the task allocation problem of multiple heterogeneous robots for optimization an unknown environment under defined constraints. The results indicate that the MILP is effective and efficient in solving problems of a practical size. The result implies that that the size of the MILP is determined by the number of tasks, and is independent of number of robots. The main drawback of using MILP techniques is that the problem is NP-hard, and takes exponential time to solve. The initial solution obtained from the first phase is utilized to decide the task performing capacities of the candidate robots. The model is then simulated by number of tasks to make it suitable for application of MILP in order to find out the optimized task allocation. But in the MILP, utilization of robots is not taken care of properly. In order to test the efficiency of the methodology an example problem with four heterogeneous robots and fifteen different tasks is worked out.

Computational results indicate that the KA is effective and efficient in solving problems of a big size as compared to other models. Future research will involve both improvements in solution methods and extensions to the current model. The large muliple knapsack prolems, despite the NP-hardness, generally are as easy to solve as ordinary 0-1 Knapsack Problems. Small instances with a reasonable $n/m$ ratio can also be handled, although large instances of the same kind are almost intractable.Thus future study should be focused on those instances, where $n/m$ is small.

The HA cannot get the result in small matrices. In HA the utilization of robots is not taken care of properly. The results prove that HA is able to be applied to the MRS and can satisfy the optimal need.

In this task allocation, a mathematical model and six efficient allocation methods are developed to support multiple-type robot acquisition in a CIM system. Five efficient methods are developed: (1) Greedy heuristic, (2) Linear Programming, (3) Mixed Integer Linear Programming, (4) Knapsack Algorithm, (5) Hungarian Algorithm, and (6) Particle Swarm Optimization. The allocation methods are implemented in MATLAB and Lingo and tested by solving different problems based on major design parameters. Computational results indicate that the greedy heuristic is significantly more effective and efficient than an exact solution algorithm in solving problems of a practical size. In this task allocation, PSO has the less optimimal solutions as compared to the other methods. The size of the LP problem is determined by the number of tasks, and is independent of the number of robots and it is not suitable for large problems.

The integration of task allocation with assembly planning in MRS adopted here generally gives emphasis on the number of robot sequences in which assembly tasks can be executed. The larger the sequences and number of parallel actions of the assembly, the more is the flexibility in assigning the assembly tasks. Each robot is characterized by its capability in terms of the number and type of joints. The analysis of the capability of the candidate robot helps select the robot for a particular task. From the different task allocation and the associated motion study the cycle time for producing an assembly is determined and is compared with other alternatives that are generated through the process. This investigation also gives an indication about the advantage of parallelism of the execution of assembly task. The total cycle time is reduced to a large extent with parallelism/simultaneity in the execution of the assembly task. In future the work can be extended to the designing of the entire assembly system.

### 6.7 Summary

An empirical study is described in the MRS that sought general guidelines for task allocation strategies. It is clearly show that the opportunistic strategy worked considerably advanced to the commitment-based strategy.

From the description of the different methods for robot selection, the features of the robot selection process that fit well with the methodology can be summarized as follows:

1. Robot selection is a multi-attribute decision making process. The method can provide a complete solution for selecting process.
2. Expertise for robot selection process is trivial and time-consuming. The method can save designers a lot of efforts to get the answers.
3. The systems can be built without too much knowledge-elicitation effort. In the system, users do not have to understand how to solve the robot selection problem.
4. Robot selection knowledge evolves over time. The method can be used in training professional in robotic design domain.
5. Finally, by acquiring robot selection new cases, the selection system can grow to reflect their company's robotic experience.

It is noticed that the methodology may provide a useful tool for designers and managers attempting to increase design quality and efficiency. It is interesting to note that the widespread use of this method is likely to lead many designers to put their knowledge into library, that is, this method may allow developers to provide intelligent robotics knowledge services and coordinate highly collaborative design activities for designers and users.

Task allocation in MRS is inferred logically in case of both mathematical based methods and greedy heuristics based method. Conventional methods produce number of alternative solutions. The conventional methods may produce single solution depending upon type of the product and the applied logic. However, there is no means to claim that the solutions obtained through application of these conventional methods are optimal from cost, time and motion perspectives. On the other hand, the ACO method produces the optimal or near optimal solution. In conventional methods, stability of the in-process assembly is checked by the experience of assembler, whereas, stability of the in-process assembly is checked by mathematical modeling in case of ACO. The fit type relationship among the components and part handling stability are not taken into consideration while deriving the assembly sequence in case of conventional methods, whereas, this very concept is incorporated in ant algorithm.

# Chapter-VII

# CONCLUSION AND SCOPE FOR FUTURE WORK

# CHAPTER 7

# Conclusion and Scope for Future Work

## 7.1 General

Multirobot facility design and planning have become increasingly important in modem production over the past decade. There is no general theory of task allocation in uncertain multi-robot domains. In Chapter 3, an attempt is made to empirically derive some guidelines for selecting task allocation strategies for MRS with implicit cooperation. The explored strategies are individualistic in that they do not involve explicit cooperation and negotiation among the robots. However, they are a part of a large class approaches that produce coherent and efficient cooperative behavior. Given the empirical nature of this work and the scope of the problem being addressed, these guidelines are necessarily incomplete, though they provide useful insight. The choice of task allocation strategy is far from trivial and that no optimal task allocation strategy exists for all domains. It can be very difficult to identify the optimal task allocation strategy even for a particular task. These results are derived through the use of a framework developed for understanding the task allocation problem, which illustrates a common approach to decomposing the problem.

## 7.2 Robot selection for MRS

In Chapter 4, a new mathematical based methodology is proposed for robot selection to help designers identify feasible robots, and then outline the most appropriate cases for smoothing robot selection process. It deals with the issues of using past experiences or cases to understand, plan for, or learn from novel situations. The results of this study will help robot workcell designers to develop a more efficient and effective method to select robots for robot applications.

The robot selection on the basis of task requirement is aimed at developng a generalized tool to combine manipulator attributes and task requirements in a comprehensive manner for relative ranking of the manipulators. The methodology developed through this work can be applied to any similar set-up. This is sure to help the designers and users in selecting the robots correctly for the intended application. In the example a total of 13 parameters are taken into account. The factors such as arm geometry, actuator, control mode, robot programming, space and time are considered in this example for better selection of robots. As per the analysis, robot selection on the basis of task requirement is the best method as compared to the other methods.

The case based method presented in section 4.4.4 is useful for helping production engineers to select robots for robotic workcell design. On the basis of these example results, it may be concluded that this method is a pragmatic approach for robot selection application. When the cases increase, the system becomes more useful for robot selection. In order to improve the performance, it is recommended that the system user should collect more cases from historical robot applications and production processes. The need to combine the product development activities with production process and robot selection information has been emphasized.

In the present work, a mathematical model and solution algorithm is developed section 4.4.3 to support robot selection and task assignment in a system employing multiple robot types. Specifically, the developed model considers selection of a proper mix of multiple type robots such that operational requirements for a given number of tasks are satisfied. Each robot is characterized by its unique fixed charge and subject to its machine time and space capacity constraints. Each task has known time and space demands for each type of robot.

## 7.3 Task assignment in MRS

In Chapter 5, a mathematical allocation model and six efficient allocation methods are developed to support multiple-type robot acquisition for MRS. Six efficient methods are developed. These are: (1) Greedy Heuristic, (2) Linear Programming, (3) Mixed Integer Linear Programming, (4) Knapsack Algorithm, (5) Hungarian Algorithm, and (6) Particle swarm optimization.

The allocation methods are implemented in MATLAB and Lingo and tested by solving different problems based on major design parameters.

The models of GH, LP, MILP, KA, HA and PSO are developed for multirobot assignment in Chapter 5. An optimization algorithm is developed using a greedy heuristic. The model is formulated as a pure 0-1 mathematical program, which is shown harder than the two-dimensional bin packing problem, a well-known NP-hard problem. Computational results indicate that the algorithm is effective and efficient in solving problems of a large size. The algorithm serves as a practical tool for planning facilities with multiple types of robots.

The result implies that the size of the linear programming problem is determined by the number of tasks, and is independent of number of robots. The model is initially formulated as a pure 0-1 mathematical program. The initial solution obtained from the first phase is utilized to decide the task performing capacities of the candidate robots. The model is then simulated by number of tasks to make it suitable for application of LP for finding out the optimized task allocation. In order to test the efficiency of the methodology an example problem with four heterogeneous robots and fifteen different tasks is worked out. Computational results indicate that the algorithm is effective and efficient in solving problems of a practical size. Since size of the LP problem is determined by the number of tasks, and is independent of the number of robots, hence it is not suitable for large problems.

A mathematical model and the solution algorithms are developed to support robot selection and task assignment in a system employing multiple robot types. Models of MILP are developed to solve the task allocation problem of multiple heterogeneous robots for optimization an unknown environment under defined constraints. The results indicate that the MILP is effective and efficient in solving problems of practical size. The result implies that that similar to the LP the size of the MILP is determined by the number of tasks, and is independent of number of robots. The main drawback of using MILP techniques is that the problem is NP-hard, and takes exponential time to solve. The initial solution obtained from the first phase is utilized to decide the task performing capacities of the candidate robots. The model is then simulated by number of tasks to make it suitable for application of MILP in order to find out the optimized task allocation. But in the MILP, utilization of robots is not taken care of properly. In order to test the efficiency of the methodology an example problem with four heterogeneous robots and fifteen different tasks is worked out. It is only feasible to use on smaller-scale problems, where there are a limited number of robots and tasks. Because the algorithm is static rather than dynamic, all information about the robots must be known *apriori* to solve the problem. However, this method is unable to adapt the schedule to dynamic changes in the agent network during execution of the schedule itself; if a change is made the entire schedule must be re-computed based on the new input data.

Computational results indicate that the KA is quite effective and efficient in solving problems of a big size as compared to other models. The large muliple knapsack prolems, despite the NP-hardness, generally are as easy to solve as ordinary 0-1 Knapsack Problems. Small instances with a reasonable $n/m$ ratio can also be handled, although large instances of the same kind are almost intractable.Thus future study should be focused on those instances, where $n/m$ is small.

The HA cannot get the result in small matrices. In HA the utilization of robots is not taken care of properly. The results prove that HA is able to be applied to the MRS and can satisfy the optimal need. PSO method balances between the global and local search can be adjusted through the inertial weight factor.

Computational results indicate that the PSO is significantly more effective and efficient than an exact solution algorithm in solving problems of a large size. In this task allocation, PSO exhibits the best value of optimal solution as compared to the other methods.

The integration of task allocation with assembly planning in MRS adopted in the work (section 5.7) generally gives emphasis on the number of robot sequences in which assembly tasks can be executed. The larger the sequences and number of parallel actions of the assembly, the more is the flexibility in assigning the tasks. Each robot is characterized by its capability in terms of the number and type of joints. The analysis of the capability of the candidate robot helps select the robot for a particular task. From the different task allocation and the associated motion study the cycle time for producing an assembly is determined and is compared with other alternatives that are generated through the process. This investigation also gives an indication about the advantage of parallelism of the execution of assembly task. The total cycle time is reduced to a large extent with parallelism/simultaneity in the execution of the assembly task.

### 7.4 Contributions

The following are the prime contributions towards the enrichment of the research work in planning and designing MRS for industrial purpose.

i. The present work addresses the issues of robot selection in a pragmatic manner and takes all the necessary and pertinent parameters into consideration in the developed model whereas the previous studies, as observed from literature, do consider only some specific parameters while modeling the robot selection process.

ii. The strategy for operation and task assignment in MRS are novel things that have been considered while conceptualizing the MRS and modeling the task assignment process. These strategies have been inbuilt in the models and hence the results are more realistic than other theoretical models reported in various literatures.

iii. All possible types of task assignments methodologies have been tried and compared. Their suitability for different size of MRS has been explained in the present work for the benefit of the designers of MRS, thereby increasing the domain of application of the developed methodology.

iv. Previous studies focus discretely on robot selection and task assignment. Realizing the strong relationship between the two, the present work takes a holistic view of both the processes and an attempt has been made to integrate the two processes.

v. In order to make the work practicable, an integrated approach to deal with a practical problem dealing with processes of task decomposition, task planning, robot selection and task assignment has been developed. The procedure has been explained in detail through example.

## 7.5 Scope for future work

Extensions to the work done in this thesis may include exploring a dynamic programming approach to the same problem that is capable of taking into account the uncertainty levels or a limited view of the agent environment, and can adapt to unexpected changes in the environment or problem domain during execution of the actual execution. Other future areas of research related to this thesis may also include dynamically re-organizing teams of agents that respond to changing objectives and environments. Future research will involve both improvements in solution methods and extensions to the current model.

Problem formulation and decomposition techniques are also introduced in this thesis, but no formula is developed to compute optimal task decompositions. Optimization techniques to determine the best way to formulate and decompose a problem for a given agent organization is another promising research avenue that may be pursued based on the work done in this thesis, although in many cases the optimal decomposition seems to vary depending on the nature of each individual problem.

Furthermore, it may be helpful to perform additional studies on the convergence rate of the different techniques relative to the number of robots and tasks in the problem, so that approximate computation times may be predicted in advance

In future the work can be extended to the designing of the entire assembly system. The research presented in this thesis has been built upon previous work in MRTA, but some questions were not adequately answered in the literature and were beyond the scope of this work. This section identifies some topics that merit further exploration.

# References

[1] J. Ferber, Multi-Agent Systems, Addison-Wesley, 1999.

[2] C. R. Kube and E. Bonabeau, Cooperative transport by ants and robots, Robotics and Autonomous Systems, vol. 30, no. 1, pp. 85-101, 2000.

[3] L. Iocchi, D. Nardi, and M. Salerno, Reactivity and deliberation: a survey on multi-robot systems, In Balancing Reactivity and Deliberationin Multi-Agent Systems (LNAI 2103), E. P. M. Hannebauer, J. Wendler, Ed. Springer, pp.9-32, 2001.

[4] A. C. Shultz and L. E. Parker, The effect of heterogeneity in teams of 100+ mobile robots, Proceedings of the 2003 NRL Workshop on Multi-Robot Systems, Eds., vol. II. Washington, DC: Kluwer Academic Publishers, 2003.

[5] K. Konolige et al., Centibots: Large scale robot teams, Proceedings of the 2003 NRL Workshop on Multi-Robot Systems, vol. II. Washington, DC, Kluwer Academic Publishers, pp.1-12, 2003.

[6] M. Vukobratovic, Scientific fundamentals of industrial robots 1: Dynamics of manipulator robots theory and applications, Springer-Verlag Publication, New York, 1982.

[7] V.P. Agrawal, V. Kohli, S. Gupta, Computer aided robot selection: the multiple attribute decision making approach, International Journal of Production Research 29 (8), pp.1629-1644, 1991.

[8] Moutaz Khouja, David E. Booth, Michael Suh and John K. Mahaney Jr, Statistical procedures for task assignment and robot selection in assembly cells, Int. J. Computer Integrated Manufacturing, Vol. 13, No. 2, 95-106, 2000.

[9] D.E.Booth, M.Khouja and Michael Hu, A Robust Multivariate Procedure for Evaluation and Selection of Industrial Robots, International Journal of Operations and Production Management, 12, 2, pp.15-24 ,1992.

[10] M. Dooner, Computer simulation to aid robot selection, in: A. Pugh (Ed.), Robotic Technology Journal, Peter Perigrinus Ltd., London, 1983.

[11] P.Y. Huang, P. Ghandforoush, Robotics procedures given for evaluating selecting robots, Industrial Engineering 16(4) , pp. 44-48, 1984.

[12] K. Madhuraj, “Computer aided robot selection, M. Tech Project Report, Department of Mechanical Engineering, IIT Delhi, 1988.

[13] R. Hinson, Robotics-environment a major factor in robot selection, Industrial Engineering 15(10), pp. 30-32, 1983.

[14] M.S. Jones, C.J. Malmborg, M.H. Agee, Decision support system used for robot selection, Industrial Engineering 17 (9) , pp. 66-73, 1985.

[15] R.P. Paul, S.Y. Nof, “Work methods measurements-a comparison between robot and human task performance”, International Journal of Production Research 17 (277), pp.1109-1121, 1979.

[16] R. Hinson, Knwoing work envelops helps in evaluating robots, Industrial Engineering 15 (7), pp.22-27, 1983.

[17] V.N. Rajan and S.Y.Nof, Minimal Precedence constraints for integrated assembly and execution planning, IEEE Transactions on Robotics and Automation, 12**,** pp.175-186, 1996.

[18] O.F.Offodile ,P.K.Lambert and R.A.Dudek. Development of a computer aided robot selection procedure (CARSP), Int J Prod Res, 25, pp.1109-12, 1987.

[19] G.H.Liang , M.J.Wang , A fuzzy multi-criteria decision-making approach for robot selection, Robotics and Computer Aided Manufacturing 10:pp.267-274, 1993.

[20] R.V.Rao, K.K.Padmanabhan, Selection, identification and comparison of industrial robots using digraph and matrix methods, Robotics and Computer Integrated Manufacturing, 22, pp.373-383, 2006.

[21] L.Zhao ,T. Yashuhiro, Genetic algorithm for robot selection and work station assignment problem Computers & Industrial Engineering , 31, pp.599-602, 1996.

[22] N. Boubekri , M.Sahoui and C. Lakrib, Development of an expert system for industrial robot selection. Comput Ind Eng, 20, pp.119-127, 1991.

[23] M.Khouja, and D.E. Booth, Fuzzy clustering procedure for evaluation and selection of industrial robots, Journal of Manufacturing Systems, 14, pp.244-251, 1995.

[24] P.P.Bhangale, V.P. Agrawal, S.K. Saha, Attribute based specification, comparison and selection of a robot, Mechanism & Machine Theory 39, pp.1345-1366, 2004.

[25] E. Ertugrul Karsak, A two-phase robot selection procedure, Production Planning & Control, Vol. 9, No. 7, pp.675-684, 1998.

[26] E.Ertugrul Karsak1 and S.Sebnem Ahiska1, A Common-Weight MCDM Framework for Decision Problems with Multiple Inputs and Outputs, Vol. 4705, Springer Berlin / Heidelberg, 2007.

[27] R.C.Arkin and T. Balch, Cooperative multiagent robotic systems, AAAI Press/The MIT Press, Cambridge, MA, pp.277-296, 1998.

[28] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, A taxonomy for swarm robots, Proc.of IEEE/RSJ International conference on Intelligent Robotics and Systems, pp.441-447, 1993.

[29] D. Goldberg and M.J. Mataric, Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks, USC Institute for Robotics and Intelligent Systems Technical Report IRIS-00-387, 2000.

[30] G. Sukthanker and L. Sycara., Team-aware robotic deminig agents for military simulation, Carnegie Mellon University, Robotics Institute Technical Report, 2000.

[31] K. Lerman and A. Gastyan , A general methodology for mathematical analysis of multi-agent systems,USC Information Sciences, Technical Report ISI-TR-529, 2001.

[32] T. Balch and R.C. Arkin, Behavior-based formation control for multirobot teams, IEEE Trans. Robotics and Automation, December, Vol. 14, pp.926-939, 1998.

[33] J. Fredslund and M.J. Mataric, A general local algorithm for robot formation, Proc.of IEEE International Symposium on Computational Intelligence for Robotics and Automation, Banff, Canada, 2001.

[34] B.P. Gerkey and M.J. Mataric, Murdoch: Publish/Subscribe Task Allocation For Heterogeneous Agents, Proc.of Fourth International Conference on Autonomous Agents, ACM Press, Barcelona, Spain, pp.203-204, 2000.

[35] R.C. Arkin, Motor schema based mobile robot navigation, International Journal of Robotics Research, Vol. 8, No. 4, pp.92-112, 1989.

[36] V.R. Lesser, A retrospective view of FA/C distributed problem solving, IEEE Trans. Systems, Man, and Cybernetics, Vol. 21, pp.1347-1362, 1991.

[37] L.E. Parker, ALLIANCE: an architecture for fault tolerant multi-robot cooperation, IEEE Trans. on Robotics and Automation, Vol. 14, No. 2, pp.220-240, 1998.

[38] L.E. Parker, On the design of behavior-based multi-robot teams, Journal of Advanced Robotics, Vol. 10, No. 6, pp.547-568, 1996.

[39] L.E. Parker, Designing control laws for cooperative agent teams, Proc.of IEEE International Conference on Robotics and Automation, pp.582-587,1993.

[40] L.E. Parker, Evaluating success in autonomous multi-robot teams: experiences from ALLIANCE architecture implementations, Journal of Theoretical and Experimental Artificial Intelligence, Vol. 13, pp.95-98, 2001.

[41] L.E. Parker, Cooperative motion control for multi-target observation, in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1591-1598, 1997.

[42] L.E. Parker, L-ALLIANCE: task-oriented multi-robot learning in behavior-based systems, Advanced Robotics, Special Issue on Selected Papers from IROS'96, Vol. 11, No.4, pp.305-322, 1997.

[43] L.E. Parker, L-ALLIANCE: A Mechanism for Adaptive Action Selection in Heterogeneous Multi-Robot Teams, ORNL/TM-13000, 1995.

[44] P. Pirjanian, Multiple objective action selection in behavior-based control, in Proc.of 6th Symposium for Intelligent Robotic Systems, Edinburgh, Scotland, July, pp.83-92, 1998.

[45] P. Pirjanian and H.I. Christensen, Satisfying action selection, Proc.of SPIE Conference on Intelligent Systems and Advanced Manufacturing, Boston, Massachusetts, November, pp.153-164,1989.

[46] P. Stone and M. Veloso, Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork, Artificial Intelligence, Vol. 110, No. 2, June, pp. 241-273, 1999.

[47] B.P. Gerkey and M.J. Mataric, Pusher-watcher: an approach to fault-tolerant tightly-coupled robot coordination, Proc.of IEEE/RSJ IRO, pp.464-469, 2001.

[48] B.P. Gerkey and M.J. Mataric, Sold!: Auction Methods for Multirobot Coordination , IEEE trans. Robotics and Automation, Vol.18,No.5, pp.758-768, 2001.

[49] M.Berhauld , H.Huang , P.Keskinocaki **,** S. Koenigi, W. Elmaghrabyi, P. Griffin and A. Kleywegd , Robot Exploration with Combinatorial Auctions, Proc. of Int.Conf. on Intelligent Robots and Systems ,vol.2, pp.1957-1962, 2003.

[50] D.P. Bertsekas, Auction algorithms for network flow problems: A tutorial introduction, Computational Optimization and Applications, pp. 7-66, 1992.

[51] A. Stentz, and M.B. Dias, A free market architecture for coordinating multiple robots, Carnegie Mellon Robotics Institute Tech Report CMU-RI-TR-99-42, December 1999.

[52] M. B.Dias, and A. Stentz, A Free Market Architecture for Distributed Control of a Multi-Robot System, Proc.of 6th International Conference on Intelligent Autonomous System, pp. 115-122, 2000.

[53] B.B. Werger and M.J. Mataric, Broadcast of local eligibility for multi-target observation, Proc.of DARS 4, pp.347-356, 2000.

[54] B. Werger and M.J. Mataric, From insect to internet: Situated control for networked robot teams, Annals of Mathematics and Artificial Intelligence, Vol. 31, No. 1-4, pp.173-198, 2001.

[55] E. Ostergaard, and M.J. Mataric, Distributed multi-robot task allocation for emergency handling, Proc.of IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, pp.821-826, 2001

[56] M. Dorigo, V. Maniezzo, and A. Colorni, The Ant System: Optimization by a Colony of Cooperating Agents, IEEE Trans. Systems, Man, and Cybernetics, Part B, vol. 26, No. 1, pp.29-41, 1996.

[57] M.J. Matric, G. S. Sukhatme and E. H. Østergaard., Multi-Robot Task Allocation in Uncertain Environments, Autonomous Robots 14, pp.255-263, 2003.

[58] Li Lian and R. Murray, Cooperative Task Planning of Multi-Robot Systems with Temporal Constraints, Proc. of International Conference on Robotics & Automation, pp.14-19, 2003.

[59] A. Andersson,M. Tenhunen and F.Ygge, Integer Programming for Combinatorial Auction Winner Determination, Proc.of Fourth International Conference on Multiagent Systems, pp.39-46, 2000.

[60] J. Guerrero and G. Oliver, Physical interference impact in multi-robot task allocation auction methods, Proc.of IEEE Workshop on Distributed Intelligent Systems, pp.19-24, 2006.

[61] I. Gungor and M.Gunes, Fuzzy multiplecriteria assignment problems for fusion: The case of Hungarian algorithm, Proc.of Third International Conference on Information Fusion, pp.8-12, 2000.

[62] L.Zu, Y.Tian, J.Fu and J.Liu,"Algorithm of task allocation based on realizing at the lowest cost in multirobot system," Proc.of Third International Conference on Machine Learning and Cybernatics, pp.152-156, 2004.

[63] L. Lin and Z. Zheng, Combinatorial bids based multi-robot task allocation method, Proc.of IEEE International Conference on Robotics and Automation Barcelona, pp.1145-1150, 2005.

[64] J. Cook and B. Han, Optimal robot selection and work station assignment for a CIM system, IEEE Trans. Robotics and Automation, pp.210-219, 1994.

[65] A. R. Mosteo and L. Montano, Simulated annealing for multi-robot hierarchical task allocation with MinMax objective, Workshop on Network Robot Systems; Toward intelligent Robotic Systems Integrated with Enviroments , IROS-2006.

[66] N. Boneschanscher, Task assignment for a small batch flexible assembly cell incorporating multiple robots, Delft University of Technology, Laboratory for Manufacturing Systems, IEEE, pp.746-750, 1990.

[67] A.Sahu and R. Tapadar, Solving the Assignment problem using Genetic. Algorithm and Simulated Annealing, IAENG International Journal of Applied Mathematics, 36:1, IJAM, 36, pp.1-7, 2007.

[68] L. Parker, “Alliance: An architecture for fault tolerant multirobot cooperation,”IEEE Trans. Robot. Automat., vol. 14, no. 2, pp. 220-240, 1998.

[69] C. Kube and H. Zhang, Stagnation recovery behaviors for collective robots, in Proc. IEEE/RSJ/GI Int. Conf. Intelligent Robots and Systems (IROS ’94), vol.3, pp. 1883-1890, 1994.

[70] B. B. Werger and M. Mataric, “Broadcast of local eligibility for multitarget observation,” in Distributed Autonomous Robotic Systems 4, Springer-Verlag, 2000, pp. 347-356.

[71] M. J. Mataric and G. Sukhatme, Task-allocation and coordination of multiple robots for planetary exploration, in 10th Int. Conf. Advanced Robotics, Buda, Hungary, August 2001, pp. 61-70.

[72] E. H. Ostergaard, M. J. Mataric, and G. S. Sukhatme, Distributed multirobot task allocation for emergency handling, Proc. of IEEE/RSJ Int. Conf. Robots and Systems (IROS ’01), pp. 821-826, 2001.

[73] P. Stone and M. Veloso, Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork, Artificial Intelligence, vol. 110, no. 2, pp. 241-273, 1999.

[74] C. Jones and M. J. Mataric, Adaptive division of labor in large-scale minimalist multi-robot systems, Proc. of IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS ’03), Las Vegas, pp. 27-31, 2003.

[75] C. V. Jones and M. J. Mataric, The use of internal state in multirobot coordination, proc. of Hawaii Int. Conf. on Computer Sciences,Waikiki, Hawaii, pp.27-32, 2004.

[76] R. G. Smith, The contract net protocol: High-level communication and control in a distributed problem solver, IEEE Trans. Comput., vol.29, no. 12, pp. 1104-1113, 1980.

[77] R. Davis and R. G. Smith, Negotiation as a metaphor for distributed problem solving, Artificial Intelligence, vol. 20, no. 1, pp. 63-109, 1983.

[78] P. Maes, Modeling adaptive autonomous agents, Artificial Life, 1(2), pp. 135-162, 1994.

[79] M.J. Mataric, G.S. Sukhatme, et al., Multirobot Task Allocation in uncertain Environment, Autonomous Robots, Vol. 14, pp.255-263, 2003.

[80] B. Gerkey and M.J.Mataric, Principled communication for dynamic multi-robot task allocation, Experimental Robotics VII, LNCIS 271, Springer-Verlag: Berlin, pp.353-362, 2001.

[81]M.B.Dias, and A.T.Stentz, A free market architecture for distributed control of a multirobot system, Proc. of the 6th International Conference on Intelligent Autonomous Systems (IAS-6), pp.115-122, 2000.

[82] D.D. Corkill, Blackboard systems, AI Expert, 6(9), pp.40-47, 1991

[83]B.Werger and M.Mataric, Broadcast of local eligibility for multi-target observation, Proc. of the 5th International Symposium on Distributed Autonomous Robotic Systems (DARS), Knoxville, TN, Oct. 4-6, pp.347-356, 2000.

[84] B.Werger, and M.Mataric, Broadcast of local eligibility for multi-target observation. Proc. of the 5th International Symposium on Distributed Autonomous Robotic Systems (DARS), Knoxville, TN, Oct. 4-6, 347-356, 2000.

[85] A. Seidmann, A. Arbel, and R.Shapira, A two-phase analytic approach to robotic system design, Robotics and Computer-Integrated Manufacturing, 12, pp.181-190, 1984.

[86] O. F. Offodile, and K. PO.Ugwu, Evaluating the effect of speed and payload on robot repeatability, International Journal of Robotics and Computer Integrated Manufacturing, 8, pp.27-33, 1991.

[87] P.P.Bhangale, V.P.Agrawal and S.K.Saha, Attribute based specification, comparison and selection of a robot, Mechanism & Machine Theory 39, pp.1345-1366, 2004.

[88] M.J.Khouja and R.L.Kumar, An options view of robot performance parameters in a dynamic environment, International Journal of Production Research, 37, pp.1243-1257, 1999.

[89] M.Bragilia and A. Petroni, Evaluating and selecting investments in industrial robots, International Journal of Production Research, 37, pp. 4157-4178,1999.

[90] A.M .Layek and L.J.Resare, Algorithm based decision support system for the concerted selection of equipment in machining/assembly cells, International Journal of Production Research, 38, pp.323–339, 2000.

[91] K.A. Dowsland and W.B. Dowsland, Packing problems, European Journal of Operational Research, vol.56, pp.2-14, 1992.

[92] Boyar et al , The Maximum Resource Bin Packing Problem, Proc. of the Fifteenth International Symposium on Fundamentals of Computation Theory, pp. 1-19,2005.

[93] Cormen et al, Introduction to algorithms, 2nd Edition, McGraw-Hill Publications, 2002.

[94] J.F.Goncalves and M.G.C.Resende, A Hybrid Heuristic for the Constrained Two-Dimensional Non-Guillotine Orthogonal Cutting Problem, AT&T Labs Research Technical Report TD-6UNQN6, 2006.

[95] Epstein et al., Two-Dimensional Packing with Conflicts, Proc. of the 16th International Symposium on Fundamentals of Computation Theory, 2007.

[96] S. Fujita and T. Hada, Two-Dimensional on-line Bin Packing Problem with Rotatable items Theoretical Computer Science, pp.939-952, 2002.

[97] Lodi et al, Recent Advances on Two-Dimensional Bin Packing Problems, Discrete Applied Mathematics, pp.379-396, 2002.

[98] R.P.Hernández and J.N. Dimopoulos, A New Heuristic for Solving the Multichoice Multidimensional Knapsack Problem, IEEE Transactions on Systems, Man, And Cybernetics-Part A: Systems and Humans, pp.1-10, 2005.

[99] Appleton et al., Robot Applications, New York: John Wiley & Sons, Inc., ch. 3, 1987.

[100] C. Morgan, Robots: Planning and Implementation. England: IFS Publications Ltd, 1984.

[101] C.A. Bererton, Multi-Robot Coordination and Competition Using Mixed Integer and Linear Programs, The Robotics Institute, Carnegie Mellon University, pp.1-183, 2004.

[102] G.B. Dantzig, Linear Programming, Operations Research, Vol. 50, No. 1, pp. 42-47, 2002.

[103] N. Atay and B.Bayazit, MILP Solution to Multi-Robot Task Allocation Problem, Department of computer science and Engineering, Washington University, 2006.

[104] R.P. Hernández and N.J. Dimopoulos, A New Heuristic for Solving the Multichoice Multidimensional Knapsack Problem, IEEE Trans. on Systems, Man, And Cybernetics-Part A: Systems And Humans, pp.1-10, 2005.

[105] S. Fujita and T. Hada, Two-Dimensional on-line Bin Packing Problem with Rotatable items, Theoretical Computer Science, pp.939-952, 2002.

[106] Li-Nan Zu, Yan-Taotian, Jia-Cai Fu,Ji-Fang Lrt, Algorithm of Task-Allocation Based on Realizing at the Lowest Cost in Multi Mobile Robot System, Proc. of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004.

[107] J. Wang, P. He and W. Coo, Study on the Hungarian algorithm for the maximum likelihood data association problem, Journal of Systems Engineering and Electronics, Vol. 18, No. 1, pp. 27-32, 2007.

[108] J. Kennedy, and R.C. Eberhart, Particle swarm optimization, Proc. of IEEE Int. Conf. on Neural Networks, IV, 1942-1 948. Piscataway, NJ: IEEE Service Center, 1995.

# APPENDICES

# Appendices

**A1: Algorithm for task allocation using Greedy Heuristics**

The following is the pseudocode of GH:
Z := oo /* Initial feasible solution upper bound*/
for k = 1 to K
begin
Initialization of decisions variables
for i= 1 to n /*For robot type k, calculate load
deviation ratios */
$d_{ik} := \left| t'_{ik} - s'_{ik} \right| / (t'_{ik} + s'_{ik})$
Reindex work stations such that *dlk* < *d2k* < ... < *rink*
for i= 1 to n /* Work Station Assignment Heuristic (WSAH) */
begin
Calculate adjusted allocation costs
for work station i
Assign work station i
end
if Z < Z then
begin
Z'--- Z
Update incumbent robot configuration
and work station assignment
end
end

## A2: Algorithm for task allocation using Linear Programming

MIN=0.475*X1+0.355*X2+0.355*X3+0.355*X4+0.786*X5+0.786*X6+0.786*X7+0.786*X8+0.786*X9+0.733*X10+0.733*X11+0.733*X12+0.733*X13+0.733*X14+0.733*X15+0.513*X16+0.384*X17+0.384*X18+0.384*X19+0.852*X20+0.852*X21+0.852*X22+0.852*X23+0.852*X24+0.795*X25+0.795*X26+0.795*X27+0.795*X28+0.795*X29+0.795*X30+0.513*X31+0.384*X32+0.384*X33+0.384*X34+0.852*X35+0.852*X36+0.852*X37+0.852*X38+0.852*X39+0.79*X40+0.79*X41+0.79*X42+0.79*X43+0.79*X44+0.79*X45+0.553*X46+0.414*X47+0.414*X48+0.414*X49+0.912*X50+0.912*X51+0.912*X52+0.912*X53+0.912*X54+0.846*X55+0.846*X56+0.846*X57+0.846*X58+0.846*X59+0.846*X60+0.633*X61+0.475*X62+0.475*X63+0.475*X64+1.038*X65+1.038*X66+1.038*X67+1.038*X68+1.038*X69+0.964*X70+0.964*X71+0.964*X72+0.964*X73+0.964*X74+0.964*X75+0.658*X76+0.5*X77+0.5*X78+0.5*X79+1.086*X80+1.086*X81+1.086*X82+1.086*X83+1.086*X84+1.015*X85+1.015*X86+1.015*X87+1.015*X88+1.015*X89+1.015*X90+0.674*X91+0.507*X92+0.507*X93+0.507*X94+1.104*X95+1.104*X96+1.104*X97+1.104*X98+1.104*X99+1.021*X100+1.021*X101+1.021*X102+1.021*X103+1.021*X104+1.021*X105+0.699*X106+0.526*X107+0.526*X108+0.526*X109+1.14*X110+1.14*X111+1.14*X112+1.14*X113+1.14*X114+1.06*X115+1.06*X116+1.06*X117+1.06*X118+1.06*X119+1.06*X120+0.759*X121+0.573*X122+0.573*X123+0.573*X124+1.2*X125+1.2*X126+1.2*X127+1.2*X128+1.2*X129+1.145*X130+1.145*X131+1.145*X132+1.145*X133+1.145*X134+1.145*X135+0.767*X136+0.579*X137+579*X138+579*X139+1.248*X140+1.248*X141+1.248*X142+1.248*X143+1.248*X144+1.156*X145+1.156*X146+1.156*X147+1.156*X148+1.156*X149+1.156*X150+0.803*X151+0.607*X152+0.607*X153+0.607*X154+1.296*X155+1.296*X156+1.296*X157+1.296*X158+1.296*X159+1.201*X160+1.201*X161+1.201*X162+1.201*X163+1.201*X164+1.201*X165+0.848*X166+0.642*X167+0.642*X168+0.642*X169+1.368*X170+1.368*X171+1.368*X172+1.368*X173+1.368*X174+1.269*X175+1.269*X176+1.269*X177+1.269*X178+1.269*X179+1.269*X180+0.848*X181+0.643*X182+0.643*X183+0.643*X184+1.362*X185+1.362*X186+1.362*X187+1.362*X188+1.362*X189+1.263*X190+1.263*X191+1.263*X192+1.263*X193+1.263*X194+1.263*X195+0.923*X196+0.703*X197+0.703*X198+0.703*X199+1.47*X200+1.47*X201+1.47*X202+1.47*X203+1.47*X204+1.359*X205+1.359*X206+1.359*X207+1.359*X208+1.359*X209+1.359*X210+0.942*X211+0.719*X212+0.719*X213+0.719*X214+1.5*X215+1.5*X216+1.5*X217+1.5*X218+1.5*X219+1.387*X220+1.387*X221+1.387*X222+1.387*X223+1.387*X224+1.387*X225;

1*X1+1*X2+1*X3+1*X4+1*X5+1*X6+1*X7+1*X8+1*X9+1*X10+1*X11+1*X12+1*X13+1*X14+1*X15=1;
1*X16+1*X17+1*X18+1*X19+1*X20+1*X21+1*X22+1*X23+1*X24+1*X25+1*X26+1*X27+1*X28+1*X29+1*X30=1;

1*X31+1*X32+1*X33+1*X34+1*X35+1*X36+1*X37+1*X38+1*X39+1*X40+1*X41+1*X42+1*X43+1*X44+1*X45=1;
1*X46+1*X47+1*X48+1*X49+1*X50+1*X51+1*X52+1*X53+1*X54+1*X55+1*X56+1*X57+1*X58+1*X59+1*X60=1;
1*X61+1*X62+1*X63+1*X64+1*X65+1*X66+1*X67+1*X68+1*X69+1*X70+1*X71+1*X72+1*X73+1*X74+1*X75=1;
1*X76+1*X77+1*X78+1*X79+1*X80+1*X81+1*X82+1*X83+1*X84+1*X85+1*X86+1*X87+1*X88+1*X89+1*X90=1;
1*X91+1*X92+1*X93+1*X94+1*X95+1*X96+1*X97+1*X98+1*X99+1*X100+1*X101+1*X102+1*X103+1*X104+1*X105=1;
X106+1*X107+1*X108+1*X109+1*X110+1*X111+1*X112+1*X113+1*X114+1*X115+1*X116+1*X117+1*X118+1*X119+1*X120=1;
1*X121+1*X122+1*X123+1*X124+1*X125+1*X126+1*X127+1*X128+1*X129+1*X130+1*X131+1*X132+1*X133+1*X134+1*X135=1;
1*X136+1*X137+1*X138+1*X139+1*X140+1*X141+1*X142+1*X143+1*X144+1*X145+1*X146+1*X147+1*X148+1*X149+1*X150=1;
1*X151+1*X152+1*X153+1*X154+1*X155+1*X156+1*X157+1*X158+1*X159+1*X160+1*X161+1*X162+1*X163+1*X164+1*X165=1;
1*X166+1*X167+1*X168+1*X169+1*X170+1*X171+1*X172+1*X173+1*X174+1*X175+1*X176+1*X177+1*X178+1*X179+1*X180=1;
1*X181+1*X182+1*X183+1*X184+1*X185+1*X186+1*X187+1*X188+1*X189+1*X190+1*X191+1*X192+1*X193+1*X194+1*X195=1;
1*X196+1*X197+1*X198+1*X199+1*X200+1*X201+1*X202+1*X203+1*X204+1*X205+1*X206+1*X207+1*X208+1*X209+1*X210=1;
X211+1*X212+1*X213+1*X214+1*X215+1*X216+1*X217+1*X218+1*X219+1*X220+1*X221+1*X222+1*X223+1*X224+1*X225=1;
1*X1+1*X16+1*X31+1*X46+1*X61+1*X76+1*X91+1*X106+1*X121+1*X136+1*X151+1*X166+1*X181+1*X196+1*X211=1;
1*X2+1*X17+1*X32+1*X47+1*X62+1*X77+1*X92+1*X107+1*X122+1*X137+1*X152+1*X167+1*X182+1*X197+1*X212=1;
1*X3+1*X18+1*X33+1*X48+1*X63+1*X78+1*X93+1*X108+1*X123+1*X138+1*X153+1*X168+1*X183+1*X198+1*X213=1;
1*X4+1*X19+1*X34+1*X49+1*X64+1*X79+1*X94+1*X109+1*X124+1*X139+1*X154+1*X169+1*X184+1*X199+1*X214=1;
1*X5+1*X20+1*X35+1*X50+1*X65+1*X80+1*X95+1*X110+1*X125+1*X140+1*X155+1*X170+1*X185+1*X200+1*X215=1;
1*X6+1*X21+1*X36+1*X51+1*X66+1*X81+1*X96+1*X111+1*X126+1*X141+1*X156+1*X171+1*X186+1*X201+1*X216=1;
1*X7+1*X22+1*X37+1*X52+1*X67+1*X82+1*X97+1*X112+1*X127+1*X142+1*X157+1*X172+1*X187+1*X202+1*X217=1;
1*X8+1*X23+1*X38+1*X53+1*X68+1*X83+1*X98+1*X113+1*X128+1*X143+1*X158+1*X173+1*X188+1*X203+1*X218=1;
1*X9+1*X24+1*X39+1*X54+1*X69+1*X84+1*X99+1*X114+1*X129+1*X144+1*X159+1*X174+1*X189+1*X204+1*X219=1;

1*X10+1*X25+1*X40+1*X55+1*X70+1*X85+1*X100+1*X115+1*X130+1*X145+1*X160+1*X175+1*X190+1*X205+1*X220=1;
1*X11+1*X26+1*X41+1*X56+1*X71+1*X86+1*X101+1*X116+1*X131+1*X146+1*X161+1*X176+1*X191+1*X206+1*X221=1;
1*X12+1*X27+1*X42+1*X57+1*X72+1*X87+1*X102+1*X117+1*X132+1*X147+1*X162+1*X177+1*X192+1*X207+1*X222=1;
1*X13+1*X28+1*X43+1*X58+1*X73+1*X88+1*X103+1*X118+1*X133+1*X148+1*X163+1*X178+1*X193+1*X208+1*X223=1;
1*X14+1*X29+1*X44+1*X59+1*X74+1*X89+1*X104+1*X119+1*X134+1*X149+1*X164+1*X179+1*X194+1*X209+1*X224=1;
1*X15+1*X30+1*X45+1*X60+1*X75+1*X90+1*X105+1*X120+1*X135+1*X150+1*X165+1*X180+1*X195+1*X210+1*X225=1;

**A3: Algorithm for task allocation using Knapsack Algorithm**

```
MODEL:
!   Robot 1 with 15 tasks;

SETS:
  ITEMS: INCLUDE, WEIGHT, RATING;
ENDSETS

DATA:
      ITEMS              WEIGHT        RATING =
         Task1              1             7.87;
         Task2              1             7.299
         Task3              1             7.299;
         Task4              1             6.8
         Task5              1             5.91
         Task6              1             5.55
         Task7              1             5.37
         Task8              1             4.878
         Task9              1             4.854
         Task10             1             4.672
         Task11             1             4.405
         Task12             1             4.405
         Task13             1             4.048
         Task14             1             3.95
         Task15             1             3.508;


  KNAPSACK_CAPACITY = 3;
ENDDATA

MAX = @SUM( ITEMS: RATING * INCLUDE);

@SUM( ITEMS: WEIGHT * INCLUDE) <=
 KNAPSACK_CAPACITY;

@FOR( ITEMS: @BIN( INCLUDE));

END

MODEL:
!   Robot 2 with 15 tasks;

SETS:
  ITEMS: INCLUDE, WEIGHT, RATING;
ENDSETS
```

```
DATA:
    ITEMS              WEIGHT      RATING =
        Task1          1           13.966
        Task2          1           12.987
        Task3          1           12.987
        Task4          1           12.004
        Task5          1           10.52
        Task6          1           9.803
        Task7           1          9.433
        Task8          1           8.695
        Task9          1           8.62
        Task10         1           8.196
        Task11          1          7.751
        Task12         1           7.751
        Task13         1           7.042
        Task14         1           6.896
        Task15         1           6.068;


  KNAPSACK_CAPACITY = 3;
ENDDATA

MAX = @SUM( ITEMS: RATING * INCLUDE);

@SUM( ITEMS: WEIGHT * INCLUDE) <=
 KNAPSACK_CAPACITY;

@FOR( ITEMS: @BIN( INCLUDE));

END
MODEL:
!   Robot 3 with 15 tasks;

SETS:
  ITEMS: INCLUDE, WEIGHT, RATING;
ENDSETS

DATA:
    ITEMS              WEIGHT      RATING =
        Task1                 1    5
        Task2                 1    4.651
        Task3                 1    4.608
        Task4                 1    4.329
        Task5                 1    3.802
        Task6                 1    3.597
        Task7                 1    3.46
```

```
        Task8                  1         3.205
        Task9                  1          3.174
        Task10                 1          3.048
        Task11                 1          2.898
        Task12                 1          2.89
        Task13                 1          2.695
        Task14                 1          2.645
        Task15                 1         2.392;


  KNAPSACK_CAPACITY = 5;
ENDDATA

MAX = @SUM( ITEMS: RATING * INCLUDE);

@SUM( ITEMS: WEIGHT * INCLUDE) <=
 KNAPSACK_CAPACITY;

@FOR( ITEMS: @BIN( INCLUDE));

END


MODEL:
!   Robot 4 with 15 tasks;

SETS:
  ITEMS: INCLUDE, WEIGHT, RATING;
ENDSETS

DATA:
    ITEMS            WEIGHT        RATING =
        Task1            1             4.651
        Task2            1             4.291
        Task3            1             4.291
        Task4            1             4.016
        Task5            1             3.521
        Task6            1             3.311
        Task7            1             3.205
        Task8            1             3.048
        Task9            1             2.932
        Task10           1             2.824
        Task11           1             2.68
        Task12           1             2.673
        Task13           1             2.487
        Task14           1             2.439
```

```
        Task15          1                    2.192;


  KNAPSACK_CAPACITY = 4;
ENDDATA

MAX = @SUM( ITEMS: RATING * INCLUDE);

@SUM( ITEMS: WEIGHT * INCLUDE) <=
 KNAPSACK_CAPACITY;

@FOR( ITEMS: @BIN( INCLUDE));

END


MODEL:
!   All Robots with 15 tasks;

SETS:
  ITEMS: INCLUDE, WEIGHT, RATING;
ENDSETS

DATA:
    ITEMS              WEIGHT       RATING =
        Task1       1             0.355
        Task2       1             0.384
        Task3       1             0.384
        Task4       1             0.553
        Task5       1             0.592
        Task6       1             0.633
        Task7       1             1.015
        Task8       1             1.021
        Task9       1             1.06
        Task10      1             1.145
        Task11      1             1.156
        Task12      1             1.296
        Task13      1             1.368
        Task14      1             1.362
        Task15      1             1.47;


  KNAPSACK_CAPACITY = 15;
ENDDATA

MAX = @SUM( ITEMS: RATING * INCLUDE);
```

```
@SUM( ITEMS: WEIGHT * INCLUDE) <=
 KNAPSACK_CAPACITY;

@FOR( ITEMS: @BIN( INCLUDE));

END
```

## A4: Algorithm for task allocation using Hungarian Algorithm

Assumption: There are n “tasks” and n “robots”.

Step 0: If necessary, convert the problem from a maximum assignment into a minimum assignment. We do this by letting C = maximum value in the assignment matrix.Replace each $c_{ij}$ with $C - c_{ij}$.

Step1: From each row subtract off the row min.

Step 2: From each column subtract off the row column min.

Step 3: Use as few lines as possible to cover all the zeros in the matrix. There is no easy rule to do this – basically trial and error.
Suppose you use k lines.

- If $k < n$, let m be the minimum uncovered number. Subtract m from every uncovered number. Add m to every number covered with two lines. Go back to the start of step 3.
- If $k = n$, goto step 4.

Step 4: Starting with the top row, work your way downwards as you make assignments. An assignment can be (uniquely) made when there is exactly one zero in a row. Once an assignment it made, delete that row and column from the matrix. If you cannot make all n assignments and all the remaining rows contain more than one zero, switch to columns. Starting with the left column, work your way rightwards as you make assignments. Iterate between row assignments and column assignments until you’ve made as many unique assignments as possible. If still haven’t made n assignments and you cannot make a unique assignment either with rows or columns, make one arbitrarily by selecting a cell with a zero in it. Then try to make unique row and/or column assignments. (See the examples below).

**A5: Algorithm for task allocation using Particle swarm optimization**

The pseudo code of the procedure is as follows

```
For each particle
   Initialize particle
END

Do
   For each particle
      Calculate fitness value
      If the fitness value is better than the best fitness value (pBest) in history
         set current value as the new pBest
   End

   Choose the particle with the best fitness value of all the particles as the gBest
   For each particle
      Calculate particle velocity according to  equation
      Update particle position according equation
   End
While maximum iterations or minimum error criteria is not attained
```

*CURRICULUM VITAE*

**NATIONAL INSTITUTE OF TECHNOLOGY**

**ROURKELA**

**PERSONAL PROFILE**

**Name** : **BIBHUTI BHUSAN CHOUDHURY**

**Father's Name** : **CHITTA RANJAN CHOUDHURY**

**Date of Birth** : 15th August 1974

**Fields of Interest** : Robotics, FMS, CAD/CAM/CIM & Soft Computing

**Phone: +91 9437166989, E-mail: bbcnit@gmail.com**

## Academic Qualification

| | |
|---|---|
| 2009 | Ph.D Dissertation Submitted, NIT Rourkela |
| 2003 | M.E (Mechanical Engg.), UCE, Burla |
| 1999 | B.E (Mechanical Engg.), Marathwada University |
| 1995 | Diploma (Mechanical Engg.) |

## Work Experience

| Employer | Date of Joining | Date of Leaving | Nature of Job |
|---|---|---|---|
| Gandhi Institute of Engg. Technology, Gunupur | 16.12.1998 | 14.9.1999 | Teaching |
| Degree Engineering College, SMIT, Berhampur | 15.9.1999 | 6.2.2004 | Teaching |
| Roland Institute of Technology, Berhampur | 8.2.2004 | 17.7.2006 | Teaching |
| NIT Rourkela (Ph.D Scholar) | 3 years | | Research |
| IGIT Sarang (An Autonomous Institute of Govt. of Orissa) | 3.3.2009 | Till date | Teaching & Research |

## PUBLICATIONS

### ***International Journals***

1. B.B.Choudhury, D.Mishra and B.B.Biswal "Task assignment and scheduling in a constrained manufacturing system using GA" International Journal of Agile Systems and Management (IJASM), Vol. 3, Nos. 1/2, 2008.
2. B.B.Biswal and B.B.Choudhury "Development and Simulation of a Task Assignment Model for Multirobot Systems", International Journal of Engineering, Volume (1): Issue (2), pp.12-23, 2007.
3. B.B.Biswal, B.B.Choudhury ,D.Mishra, P.Dash and "An Overview and Comparison of Four Sequence Generating Methods for Robotic Assembly", **In press** in the International Journal of Manufacturing Technology and Management.(IJMTM).
4. B.B.Choudhury and B.B.Biswal "Efficient and Optimized Allocation of Tasks in Multirobot Workcell", International Journal of Computer Science and Management System (IJCSMS), pp.27-36, 2009.
5. B.B.Choudhury and B.B.Biswal "Alternative Methods for Multirobot Task Allocation" Journal of Advanced Manufacturing Systems, World Scientific, vol-8(2), pp.163-1762009.

### ***Research Papers in Conferences***

### ***International Conferences***

1. B.B.Biswal and B.B.Choudhury "Cooperative Task Planning of Multi-Robot Systems", Proc. of 24th International Symposium on Automation and Robotics in Construction (ISARC 2007) Construction & Automation Group, IIT Madras, pp.211-216, 2007.
2. B.B.Choudhury and B.B.Biswal "An Optimized Multirobot Task Allocation" Proceedings of First International Conference on Emerging Trends in Engineering and Technology, IEEE Computer Society, pp-320-325., 2008.
3. B.B.Choudhury and B.B. Biswal "Task Allocation Methodologies for Multi-Robot Systems" Proc. of IEEE CICCRA, N.I.T, Rourkela, pp.99-106, 2008.
4. B.B.Choudhury and B.B.Biswal "An Improved Method for Optimization of Multirobot Task Allocation", 2nd International and 23rd AIMTDR Conference is being organized by IIT Madras, pp.935-940, 2008.
5. B.B.Choudhury and B.B.Biswal "Selection of Robots for Task Assignment", Proc. Of IEEE International Conference on Advanced Computer Control, Singapore, IEEE Computer Society, pp-255-259, 2009.
6. B.B.Choudhury B.B.Biswal and R.N.Mahapatra "Attribute-based Ranking and Selection of Robots for Task Assignment" Proc. of IEEE International Conference on Advanced Computer Control, Singapore, IEEE Computer Society, pp-459-463, 2009.

12. B.B.Choudhury, B.B.Biswal and R.N.Mahapatra "Attribute-based Relative Ranking of Robot for Task Assignment", Proc. of IEEE International Conference on

Computer Engineering and Technology, Singapore , IEEE Computer Society, pp.32-36, 2009.

13. B.B.Choudhury, B.B.Biswal and D.Mishra "Development of Optimal Strategies for Task Assignment in Multirobot Systems" Proc. of IEEE International Advanced Conference on Recent Advance Computing Conference, Thapar University, pp.1130-1135, 2009.

14. S.Sharma, B.B.Biswal, P.Das and B.B.Choudhury "Generation of Optimized Robotic Assembly Sequence using Ant Colony Optimization", Proc. of IEEE CASE, Washington DC, USA, pp. 894-899, 2008.

7. B.B.Choudhury, D.Mishra and B.B.Biswal "Scheduling of Flexible Manufacturing System using Genetic Algorithm" Proc of International conference on Advanced Design and Manufacturing, Madurai, pp.600-607, 2007.

8. B.B.Choudhury and B.B. Biswal "Optimal Task Allocation in a Multirobot Environment using Capability Indices" Proc. of 1st International Conference of Advanced Computing ,Chikhli, Organized by ACM,pp.1-5, 2008.

9. D.Mishra, B.B.Biswal and B.B.Choudhury "A genetic algorithm approach for scheduling flexible manufacturing system with setup constraint" Proc. of International Conference on Computer Aided Engineering (CAE 2007) Department of Mechanical Engineering, I.I.T. Madras,pp.843-851.

10. N.Jena, B.B.Biswal and B.B.Choudhury "A Hybrid Evolutionary Algorithm for Flexible Job-Shop Scheduling Problem " Proc. of International conference on Advanced Design and Manufacturing ,Madurai,pp.613-619, 2007.

15. B.B.Biswal and B.B.Choudhury "Task Allocation Stratigies for Multi-Robot Systems" Proc. of International Conference on Modeling and Simulation, Coimbatore, 2007.

## ***National Conferences***

1. B.B.Choudhury and B.B.Biswal "Appropriate Solutions to Task Allocation Problems in Multi-robot Systems", National Conference on Mechanism Science and Technologies: from Theory to Application", PP.54-62, NIT Hamirpur, pp.54-62, 2008.

2. B.B.Choudhury, B.B.Biswal and D.Mishra "Task Allocation in Multi-Robot Environments" National Conference on Recent Advances in Mechanical Engineering, pp.78-82, NIT Silchar, 2008.

## ***Journals Communicated***

1. B.B.Choudhury, B.B.Biswal and S.Sharma, "Optimal Task Allocation in a Multirobot Environment Using Cost Indices", Communicated to Journal of Mechanical Engineering, ICFAI Press.

2. B.B.Choudhury S.Sharma and B.B.Biswal "Integrated Multi-Robotic Assembly Planning for Improved Productivity" Communicated to International Journal of Computational Vision and Robotics (IJCVR)

3. B.B.Choudhury ,S.Sharma and B.B.Biswal “A Strategic approach to integrated Product Design for Small to Medium Sized Enterprises” Communicated to Journal of Interscience Management Review.

4. B.B.Choudhury S.Sharma and B.B.Biswal “An Integrated Approach to Optimized Assembly Planning and Task Assignment in Multi-Robotic Cell” Communicated to International Journal of Computer Science and Management System (IJCSMS).

6. B.B.Choudhury S.Sharma and B.B.Biswal “An Optimized Multirobotic Assembly System for Industrial Applications” Communicated to International Journal of Advaced Robotic Systems.